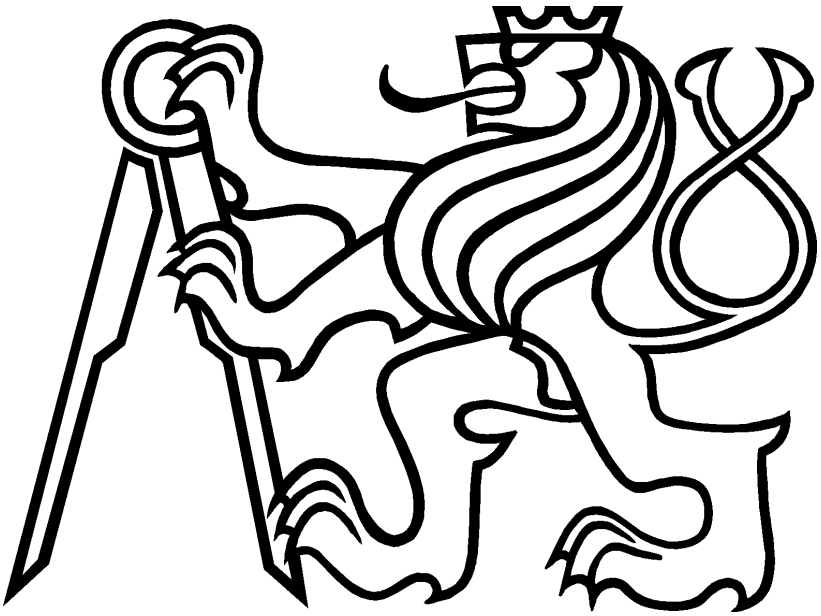
CZECH TECHNICAL UNIVERSITY IN PRAGUE

Faculty of Electrical Engineering

Diploma thesis



Bc. Pavel Kovář

User interface for the assembly line worker

Cybernetics and Robotics

Thesis supervisor: **Ing. Pavel Vrba, Ph.D.**

Declaration

I hereby declare that I have completed this thesis independently and that I have listed all used information sources in accordance with Methodical instruction about ethical principles in the preparation of university theses.

In Prague on..... .....

Acknowledgements

I would like to thank my supervisor Ing. Pavel Vrba, Ph.D. for his advices, guidance and valuable constructive criticism during this project. I would also like to thank to colleagues participating on ARUM project.

My greatest thanks go to my family, my brother-in-law and to my girlfriend for their support and encouragement during writing of this thesis.

Abstrakt

Práce popisuje proces návrhu, implementace a testování nástroje určeného pro orgazaci práce pracovníka montážní linky. Tento nástroj pracovníkovi umožňuje spravovat jeho dílčí úkoly v rámci komplexního výrobního plánu. Pokud se vyskytne nestandartní situace během výrobního procesu, je pracovníkem v reálném čase hlášena pomocí této aplikace vyšším stupňům řízení. Vyvíjené uživatelské rozhranní je součástí komplexního softwarového řešení, které cílí na optimalizaci náběhu sériové výroby konečného produktu. Konečné softwarové řešení je definované v rámci Evropského projektu ARUM (Adaptivní management produkce).

Abstract

This thesis describes the process of design, implementation and testing of tool used for assembly line worker activity management. With use of this tool the worker is allowed to monitor and manage jobs planned for himself within the scope of complex product manufacturing plan. Worker is also allowed to report issues to responsible authorities during working procedures. The tool is a part of a larger ICT system which aim is to improve ramp up period of small lot product manufacturing. The superior ICT system is defined within European project ARUM (Adaptive production management).

Czech Technical University in Prague
Faculty of Electrical Engineering

Department of Cybernetics

DIPLOMA THESIS ASSIGNMENT

Student: Bc. Pavel K o v á ř

Study programme: Cybernetics and Robotics

Specialisation: Robotics

Title of Diploma Thesis: User Interface for the Assembly Line Worker

Guidelines:

The goal of the work is to develop a user interface for mobile phone/tablet on the Android platform for the assembly line worker support. The user interface displays the list of daily jobs assigned to the worker and allows him/her to report on the work progress (job start, job end, and reporting the non-conformity or missing part). The user interface will be scalable for different screen resolutions (4.5 – 10 inches) and will communicate over the REST API with the software infrastructure developed within the ARUM project. The interface will support the role of the work inspector in the Iacobucci use-case. The tests will include the demonstration of the interaction between the user interface of the worker and of the inspector in case of handling the non-conformity.

Bibliography/Sources:

[1] Jim Webber, Savas Parastatidis, Ian Robinson, REST in Practice: Hypermedia and Systems Architecture. O'Reilly Media, 2010.
[2] Joseph Annuzzi, Lauren Darcey, Shane Conder: Introduction to Android Application Development: Android Essentials (4th Edition) (Developer's Library), Addison-Wesley, 2013.

Diploma Thesis Supervisor: Ing. Pavel Vrba, Ph.D.

Valid until: the end of the winter semester of academic year 2015/2016

L.S.

doc. Dr. Ing. Jan Kybic
Head of Department

prof. Ing. Pavel Ripka, CSc.
Dean

Prague, September 15, 2014

České vysoké učení technické v Praze
Fakulta elektrotechnická

Katedra kybernetiky

ZADÁNÍ DIPLOMOVÉ PRÁCE

Student: Bc. Pavel K o v á ř

Studijní program: Kybernetika a robotika (magisterský)

Obor: Robotika

Název tématu: Uživatelské rozhraní pro pracovníka montážní linky

Pokyny pro vypracování:

Cílem práce je vytvoření uživatelského rozhraní pro mobilní telefon/tablet na platformě Android pro podporu činností pracovníka montážní linky. Uživatelské rozhraní zobrazuje seznam denních úkolů přiřazených pracovníkovi a umožňuje mu reportovat o postupu prací (zahájení úkolu, ukončení úkolu, a nahlášení defektu či chybějící součástky). Rozhraní bude škálovatelné pro různé velikosti obrazovky (4,5 – 10 palců) a bude komunikovat přes rozhraní REST se softwarovou infrastrukturou vyvíjenou v projektu ARUM. Rozhraní bude podporovat roli kontrolora postupu prací pro scénář Iacobucci a součástí testů bude demonstrace interakce mezi uživatelským rozhraním pracovníka a kontrolora v případě reportování nekonformity.

Seznam odborné literatury:

[1] Jim Webber, Savas Parastatidis, Ian Robinson, REST in Practice: Hypermedia and Systems Architecture. O'Reilly Media, 2010.
[2] Joseph Annuzzi, Lauren Darcey, Shane Conder: Introduction to Android Application Development: Android Essentials (4th Edition) (Developer's Library), Addison-Wesley, 2013.

Vedoucí diplomové práce: Ing. Pavel Vrba, Ph.D.

Platnost zadání: do konce zimního semestru 2015/2016

L.S.

doc. Dr. Ing. Jan Kybic
vedoucí katedry

prof. Ing. Pavel Ripka, CSc.
děkan

V Praze dne 15. 9. 2014

Contents

List of Figures

List of Tables

List of Appendices

1 Introduction

This chapter presents aims, motivations and brief description of this thesis and corresponding European project. The outline of the whole thesis may be found at the end of this chapter.

1.1 Aim

The aim of this thesis is the design, the implementation and the testing of a tool used for the assembly line worker's activity management. The tool is partially universal and it was designed for two different production schemes. Partially universal means that with some level of effort it may be used for different assembly manufacturing plants. The first scheme is a production support of aeroplanes assembly in Airbus company. The second use case is a production of coffee makers in Iacobucci company. The tool allows a worker to monitor and manage jobs planned for himself within the scope of a complex production plan. It shows list of jobs to be done within the shift, reports the start, the progress and the end of the jobs. It displays complementary informations as blueprints, work instructions and required resources and tools about particular jobs. Moreover it enables the worker to report issues to responsible superiors during work.

The presented worker console is a constituent part of a complex software solution developed within the European ARUM project, which is aimed at the optimization of assembly line processes of complex and highly customised products during the ramp-up period.

1.2 ARUM project

Collaborative research project ARUM (Adaptive Production Management) was raised within European Unions 7^{th} Framework Programme [1] and is supported by the European Research Initiative "Factory of the future."[2]. The project is scheduled from 1st September 2012 to 1st October 2015 and work is divided between 13 partners from European Union and one partner from Russian Federation. ARUM is registered with project number 314056 and there is dedicated budget of €11,6 million with a contribution of €8,5 million from European Commission. The project is more closely introduced in section 2.1.

1.3 Motivation

The motivation of the ARUM project is a highly challenging ramp-up phase of production of complex and customized products, such as airplanes. The ramp-up production faces multiple disruptions such as missing parts and other resources, non-conform parts, late change requests, etc. This makes the operational day-to-day scheduling extremely difficult. The existing planning and scheduling solutions, often based on SAP, do not provide

a real-time adaptation so even such high tech companies like Airbus still rely on the pen and pencil, spreadsheet applications (like Excel) and the experience of line managers to create the daily schedules.

One of the main components of the ARUM is a scheduler which matches the resources to the particular jobs and computes the starts and the ends of the jobs. The console developed within scope of this thesis is a tool to support the workers to know when and on which job they should work and includes the above mentioned functionalities.

1.4 Outline

The structure of this thesis is following.

At the beginning of the chapter 2, there is described the whole ARUM project, the motivation to this project and the proposed architecture of the project solution. Further there is the analysis of industrial partner's manufacturing processes in section 2.2. The functional and non-functional requirements on developed tool is specified afterwards in section 2.3. At the end of the chapter, the situation on the market with handheld devices is described.

The chapter 3 contains description of the tool design process. Selection of targeted software platform and particular devices are selected in section 3.1. The UI design and mock-ups specifying the real use-cases are introduced in section 3.2. There is proposed architecture of the developed tool in section 3.3. The representation of information in the target device is hinted in section 3.4.

The implementation structure is briefly described in short chapter 4.

The usability testing of the Worker console and the result of the testing was documented in chapter 5.

There is displayed the set of real screen shots of the application, in chapter 6. The diagrams summarising it's functionality are present as well.

The outcome of this thesis is summarised in chapter 7. The future work on the developed tool is mentioned afterwards.

The appendices important in the context of this thesis are placed at the very end of the thesis.

2 Analysis

The ARUM project, state of the art of industrial partners, functional and non-functional requirements of the resulting tool are analysed in this chapter. The resulting specifications of the developed tool will be used as basis for it's design and development including a definition of suitable target devices and platform.

2.1 ARUM project

The ARUM project was briefly introduced in chapter 1. The general objectives, most challenging issues, aims and architecture of proposed solution are presented in this section.

2.1.1 General objectives

As it is described in the specification of project [3] and [4], the main objective of the project is the improvement of operational and economic performance of production ramp-up period by developing of new strategies, approaches and novel ICT solution.

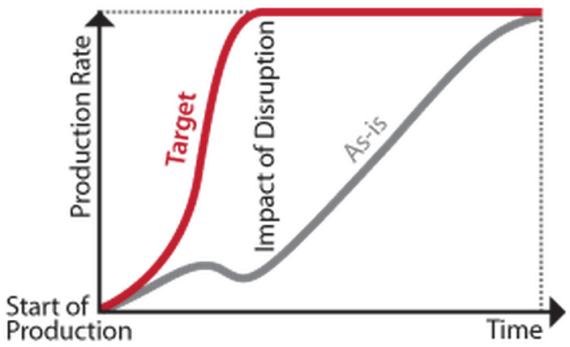


Figure 1: Ramp-up phase of production [4]

This solution would significantly improve the mentioned performances in case of a small lot or highly customised production.

Developed strategies and outcoming novel ICT system should successfully deal with challenges described in following section.

2.1.2 Challenges

- Minimization of product's immaturity and production disruptions despite market pressure.

- Speed-up of ramp-up period for complex and highly customized products within small production series.
- Improvement of planning and scheduling of automation processes.
- Application of ICT which controls manufacturing systems.
- Improvement of engineering integration to production (horizontal integration).
- Integration of enterprise ICT to shop floor level (vertical integration).

2.1.3 Ramp-up phase

Ramp-up phase of complex product's construction brings many issues that are necessary to be identified before the specification of approaches.

Complexity of product issues: In case of complex products it is impossible to teach workers how to assemble the whole product on their own. The reason is that no worker has a capacity to hold in memory every detail about the product. The second argument is that production time within this approach would be unacceptable. It is not necessary to mention that in case of larger products it would lead to unacceptable requirements on time and finances.

Safety issues: Team work on product manufacturing is logical consequence. On the other hand it requires greater managerial supervision. Quality control and verification of the work results are necessary to ensure reliability and safety of the product. Extreme emphasis is requested in case of failures threatening a human life. In this case standardization of all processes and certification of product is essential. Therefore every detail of each process has to be monitored and recorded.

Ramp-up issues: Complexity and safety issues are troublesome but may be easy resolved when producer perfectly knows a manufacturing routine.

Different issues appear when production is in prototyping or in ramp-up phase of new product. In comparison with established manufacturing the ramp-up phase is significantly affected by unpredictable disturbances such as:

- design faults
- manufacturing or supply delays
- material defects
- insufficient production capacities

- last-minute engineering or customer changes

These disturbances negatively affect production time and financial costs of the whole product. The arisen on-costs are unavoidable for each new product. The mass production projects enable to spread the costs into piece price of the product, whereas in a small lot project it is not possible. A need of production efficiency and optimised resource utilization is therefore much higher. Almost real time response on the unexpected disturbances is required for minimization of the unexpected costs.

Mentioned issues may be solved by set of new methodologies and novel event-oriented ICT system that helps resolve problems shortly after their occurrence.

2.1.4 Specification of user roles in ARUM

Each industrial partner operate with different roles of their employees. According to documents [5], [6], the roles from AIB and IHF were mapped to new set of roles within ARUM project. Table 1 shows the roles defined in the ARUM project and typical operations these roles are perfroming on a daily basis.

Functions and responsibilities important for functionality of the developed worker console will be dealt with in greater detail in section 2.3.

ARUM role	Function
Worker	- executes assigned tasks - report informations on unexpected events, delays, non-conformities to system or informs the **Team Leader** or **Station Manager** - updates information about task execution
Team Leader	- is responsible for workers team and task assignment - monitors execution of tasks, resolve minor problems and reports to **Station Manager** or **Production/Planning Manager**
Warehouse Manager	- browses list of required parts, checks availability - reports to **Production/Planning Manager**
Station Manager	- is responsible for the station performance - monitors the overall station and execution of the tasks - report to **Production Manager**
Process Manager	- is responsible for specifying the technological (manufacturing) process - is responsible for the processing of the events that happen during execution - update technological process documentation - log process changes due to disturbing events
Planning/ Production Manager	- is responsible for execution of the orders in the factory - process orders from customer support department or directly from customer - handles delays and unexpected delays to minimize losses
System administrator	- is responsible for installing of system modules and giving support to the system health and performance - is not directly involved in production

Table 1: Functions and responsibilities of ARUM roles

2.1.5 ARUM architecture

As it is described in papers [7],[8] the architecture is based on a fusion of different development approaches. The first approach include use of Multi-Agent Systems (MAS). The second one is based on use of Service Oriented Architecture (SOA). Nowadays use of intelligent agents in industrial environment is growing. Reason may be advantages coming from distributed character. (Decomposition of computational power, creation of autonomous interacting components, efficient utilization of information resources, flexibility, adaptability and so on). The SOA is pattern describing how to create modern, modular and distributed software, which is easily changeable. It is composed from reusable services. The intent

of this interconnection is encapsulation of multi-agent based functionalities to intelligent services. Advantage of this architecture is, that modular parts may be easily replaced or complemented.

The general architecture of the ARUM is shown in figure 2. Communication backbone of the architecture is Enterprise Service Bus (ESB) which provides loose coupling of components connected to bus. The ESB and components placed on it are briefly described in subsection below.

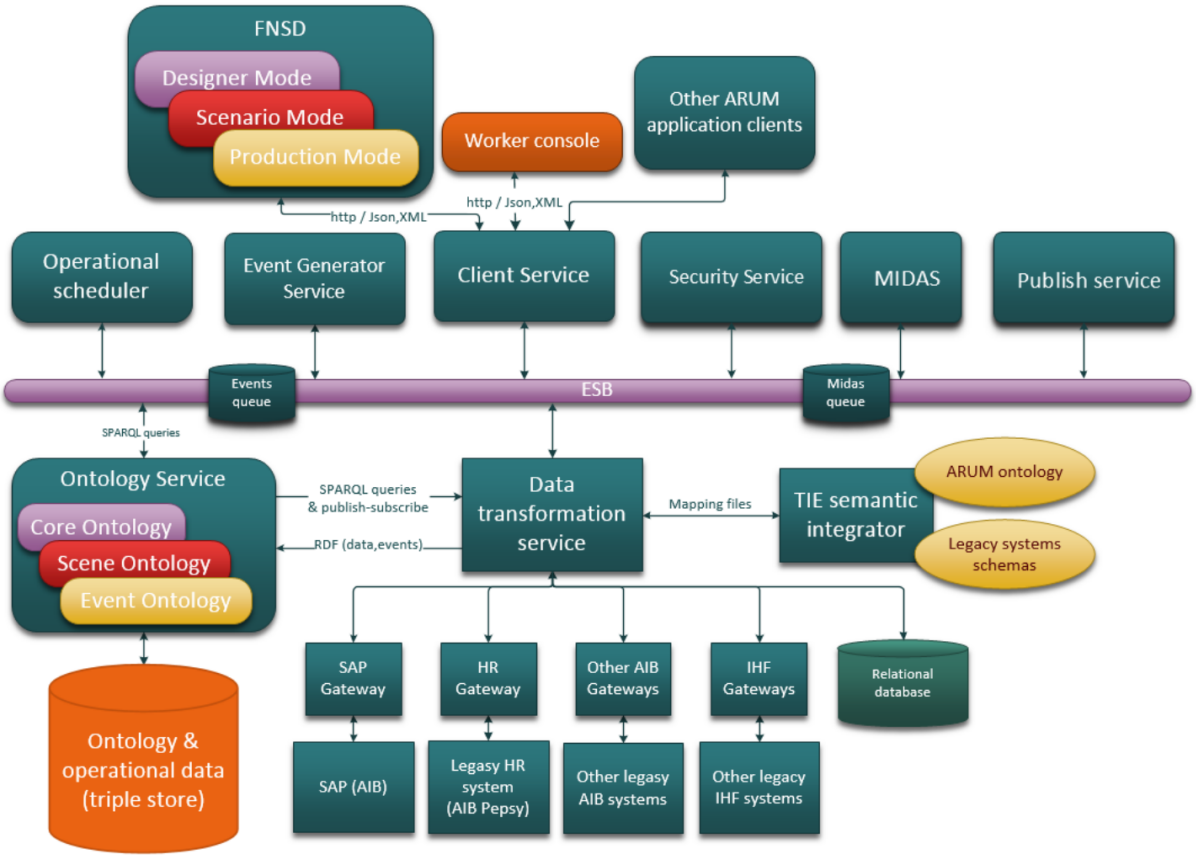


Figure 2: ARUM architecture scheme [9]

ESB According to [10], [11] Enterprise service bus (ESB) is a software architecture, which enables mutual communication between heterogeneous software applications in enterprise environment. Interaction between services connected to ESB is done with use of messages. ESB itself takes responsibility of message delivering to desired addresser. Among others, message and event queues are implemented as a way of persistent message delivering in ARUM ESB.

Description of components

- **Operational scheduler** - is a computational core of ARUM project. It schedules list of jobs and work-orders and allocates workers and time intervals to dedicated jobs.

  The scheduler consists of set of agents where each one is in charge of scheduling on particular workstation. The scheduling itself is based on constrain programming and it provides ordered list of jobs. In manufacturing process some jobs may be dependant on finishing of other ones. It is necessary to do jobs in correct order to preserve safety of final product. With help of constrain programming algorithms the mentioned jobs are assigned to persons with required skills. The other resources like parts, tools, workspace are assigned in the same way.

  The scheduling is based on a multi-agent systems because it simulates behaviour similar to human decision making including the use of heuristics. These are used because state space of scheduling issue is so complex that it is not possible to find the globally optimal solution by exploring of the whole state space in an acceptable time frame.

  The last phase of scheduling is assigning of jobs to exact timeslots and to exact persons with proper skills.

- **Event generator service** - is a service that simulates events that may happen during production. It is statistical tool which helps to estimate the real event occurrence.

- **Client service** - is interchange gateway between Enterprise Service Bus (ESB) and ARUM clients e.g. FNSD UI, Worker console, etc. The http requests from side of RESTful interface (from clients) to ARUM's other components connected with ESB service are handled by the Client service. This client is important for the functionality of FNSD UI and Worker console, because it processes and provides necessary data to them. The complete description of this web service interface is out of scope of this thesis, but it was implemented according to a architectural pattern (Representational State Transfer -REST) described in the disertation of R.T. Fielding [12]. The transfer of the objects trough this interface is done after their conversion into JSON notation and back.

- **FNSD user interface** - is an end user application used by Production Managers and other managerial staff. FNSD client consists of three different tools wrapped in single graphical user interface. Concept of Designer mode tool and Production mode tool is described in master thesis of Ondřej Harcuba [13]. Design of Scenario mode tool is described in bachelor thesis of Luis Moreno [9].

  Designer mode tool allows the Process Manager to create or edit the assembly plans of the new product or product that is actually being manufactured. The reason of changes may be last-minute modifications from customer or a reaction on manufacturing disturbances.

Production mode tool provides the Station Managers or the Production/Planning Managers with up-to date information about production status on their single working station or on the set of surrounding stations. Summary information describing state of production on the stations are reported in a form of Key Performance Indicators (KPIs). Some of the KPIs also summarize quality of calculated manufacturing plans.

Scenario mode tool is used by the Production/Planning Manager for simulation of alternative scenes. In case of need this alternative scenes may be applied to reality in the production. The scene involves several working stations, working flow trough them, resources in stations and others. The manipulation with scenes, scheduling within scenes, simulation of potential events by the Production/Planning Manager is known as what-if game. The desired result of this discipline is optimal setting of production and manufacturing plan with minimal risk of financial ant time losses.

- **Worker console** - is an end user application used by assembly line workers. The design and the implementation of this console is the main scope of this thesis. The console provides up-to-date information about jobs dedicated to the worker within applied production schedule. Whenever new schedule is applied the information about jobs are updated. The reports on the performed work on particular jobs are generated with a help of this console. The communication with responsible superiors in case of unexpected situations (delays, manufacturing faults, etc.) are significantly shortened.

  The application is also customised for the use of the Team Leader. The Team Leader is the first person who is called in when manufacturing disturbance occurs. He is the responsible person who can resolve some minor complications and it is not necessary to summon higher authority afterwards.

- **Other ARUM clients** - cover the set of unmentioned specific clients. Example of this client may be warehouse console or managerial mobile console.

- **Security service** - handles user authentication within ESB or Client service.

- **MIDAS** - is a component, which provides time estimations of troubleshooting for occurred disturbance events with respect to character of events. MIDAS is heterogeneous multi-agent system (Eve - see acticle [14]) based on web-based agent platform and using key based learning for fulfilling of the goals. It helps managers with making of decisions to resolve manufacturing discontinuities.

- **Publish service** handles incoming messages from ARUM components.

- **Ontology service and triple store** - is main storage ensuring persistent and explicit representation of information.

- **Data transformation service** - converts data from the legacy system's format (XLS, RDB, XML) to the RDF format and provides them to triple store. The transformation process is defined with the use of the TIE semantic integrator.

- **Legacy systems** are used for different purposes such as HR system, accounting system, higher managerial tools and other Airbus/Iacobucci legacy systems.
- **Relational database** is a temporary storage of manufacturing information in Airbus case. Data from this storage will be transferred to main storage in future.
- **TIE semantic integrator** - is a graphical tool used for mapping of the data schemes. In this tool the user defines the rules which help to convert data from one data type to the other ones. The conversion is executed by the mentioned transformation service. The conversion is possible for following data formats:(Excel format - XLS, Ontology format - OWL, database structure format - RDB, Extensible markup language - XML)

2.2 Industrial partner's manufacturing processes

According to the description in document [5], the manufacturing processes in Airbus and Iacobucci differ in character of products and in assembly methodologies. The subsections below describe differences and common characteristics, which are reflected in design of ARUM solution.

2.2.1 Airbus

Airbus company is separated division of international corporation Airbus Group (formerly EADS - European Aeronautic Defence and Space Company) and produces civil transport airplanes. The company is one of the leaders in aviation industry and offers 14 different plane types to over 400 customers. According to document [15] Airbus received over 1000 new orders during the year 2014. Moreover, the company has over 7000 others orders from previous years. The orders differs in type of planes and in number of ordered aeroplanes. The counts moves from 1 to 100 planes of the same type within one order. Lesser number in the order is more common and the practice is that even planes in one order differs in installed equipment. For ARUM project an A350 long-distance aircraft has been picked from the Airbus portfolio. The reason is that the A350 production is currently at the beginning of the ramp-up phase with first airplane delivered to customer at the end of 2014.

Assembly organization The aircraft assembly is organized as a line of separated workstations, where each one is equipped with a set of resources like tools, components, workers with proper skills and production manager gathered in resource pools. For six workstations there are three resource pools. It means that trunk of plane goes through all workstations during assembly process and spends limited time in each one. Required components and pre-assembled parts are provided by suppliers. Because of high value and size of parts, suppliers provide stations with goods in small bundles just-in-time.

The main task of production manager on each station is to keep station production running smoothly and with a maximised resource utilization. It should be achieved despite unpredictable events like material shortages, manufacturing faults, illness of workers and so on.

Airbus uses SAP (ERP system - Enterprise Resource Planning), which above other processes all purchase orders with delivery date to the customer. Based on this production plan is generated for single workstation down to work-order level. Work-orders are groups of single jobs that depend on each other. This production plan is provided to station manager. The issue is that this plan is more or less guideline with start and end times of work-orders because it does not reflect actual availability of resources. Particular jobs within work-orders are assigned to concrete Workers by Station manager. He does this routine with help of pen, paper and spreadsheet editor.

Each Worker gets printed list of jobs from associated work-orders dedicated to them at the beginning of their shift. The work-orders are marked in SAP as opened when they are printed by Station manager. The items in the printed list contain description of jobs the worker should do within shift and they also contain bar codes that should be scanned with central terminal after finishing of tasks. Scanning of the bar codes implies successful finishing of jobs and when all jobs from work-order are completed it marks work-orders as closed in SAP. The correctness and accomplishment of tasks during assembly process is controlled this way. It must be remarked, that the workers do not scan their bar codes instantly after accomplishment of the tasks but they usually do that after completing of greater set of tasks. The reason is that it is time consuming. However it would be necessary to actualise state of tasks instantly in case that dynamical rescheduling should be applied.

A simplified description of the assembly process and handling of the non-conformities is shown in figure 3.

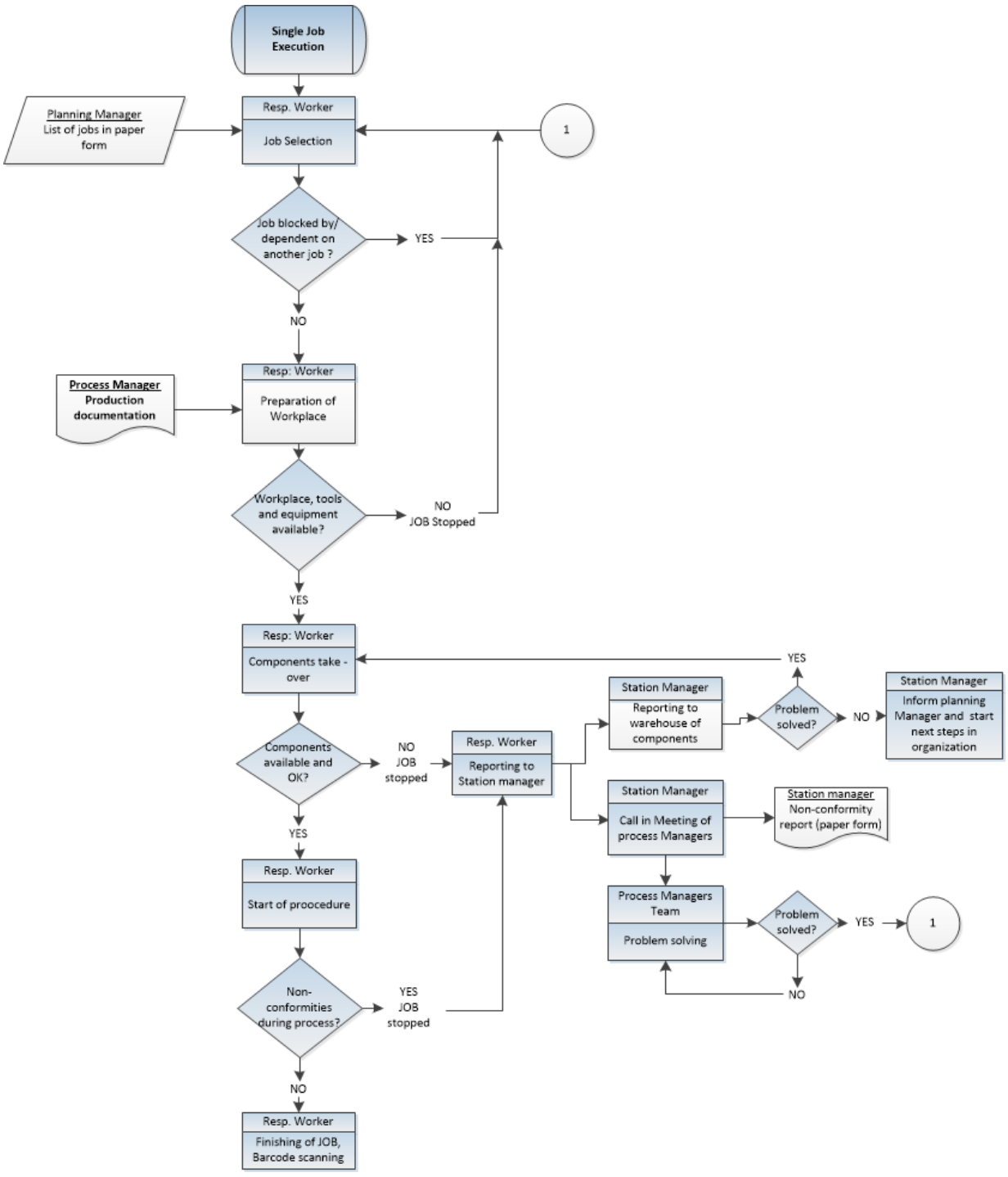


Figure 3: Schematic job workflow in AIB

2.2.2 Iacobucci

Iacobucci IHF Aerospace is Italian company producing certified coffee making automates, espresso makers or trash compactors mainly used in aerospace vehicles. Iacobucci products are widely used in Airbus planes and beside that the products are assembled into aeroplanes from others producers and into luxury yachts.

Assembly organization In contrast with Airbus use-case the IHF production assembly lines run in parallel. Workstations within lines are specialised on construction of particular products. The workload on each workstation is shared between two workers and they are able to work on multiple units in the same time. Each workstation produces five or six units of coffee makers per week. The quality is checked by the Operative Area Responsible (RAO), who is a worker with extended competence. He decides if it is necessary to call responsible superior in case of nonconformity detected during quality check or when unexpected disturbance occurs. Unlike Airbus use-case, where quality check of finished jobs is part of working procedure is planned by Station manager and performed by authorised Worker, in Iacobucci the RAO is called in to do quality check when finished.

A simplified description of the assembly process and handling of the non-conformities is shown in figure 4.

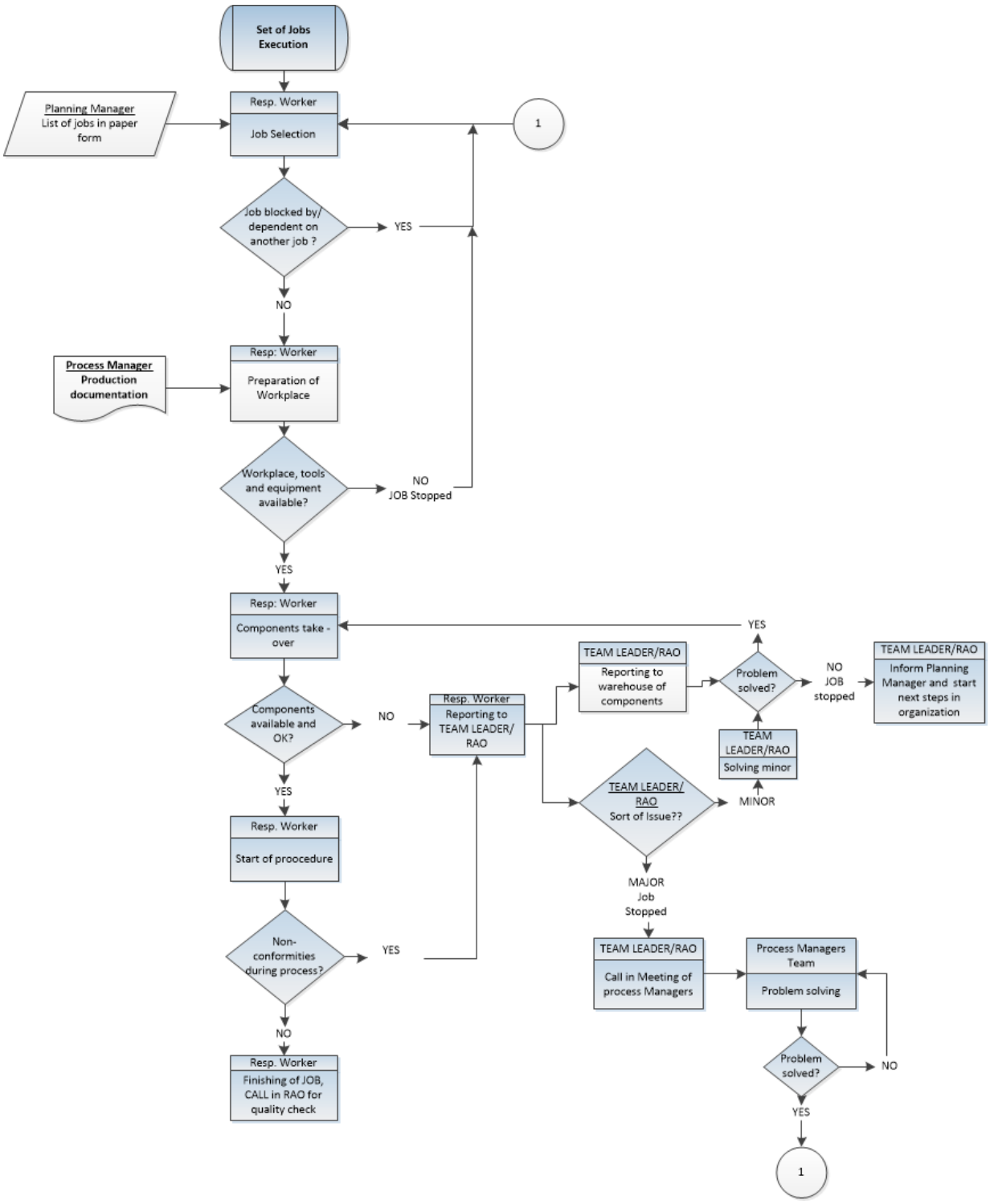


Figure 4: Schematic job workflow in IHF

2.2.3 Common characteristics

Both companies plan production fur working stations on a daily basis. As an input of planning procedure a large amount of information is necessary as listed below.

- Duration of jobs.
- Required skills for job's completion.
- Place of job's execution.
- Capacity of working place.
- Required tools and resources - defined in bill of material.
- List of tools and resources within workstation.
- Available workers within working shift and their skills - provided by HR system.

The result or output of planning procedure is a production schedule that assigns available resources, tools and workers with proper skills to particular jobs in particular time.

2.3 Worker console requirements - analysis output

The worker console users should be able to execute operations described in document [6]. The versatility of ARUM solution requires that some roles of employees may overlap. For example Operative Area Responsible (RAO) in Iacobucci use case may stand for functionality of Worker, Team Leader and sometimes he has competence to do minor decisions in role of Production Manager. The situation is different in case of Airbus, where Team Leader is not defined at all.

For better understanding the following table shows the roles and functions/responsibilities of main users of worker console enhancing the ARUM roles shown in table 1. The developed product is mainly targeted to Worker and Team leader. The System administrator will be responsible for correct set-up of this console and integration to company's internal network.

ARUM role	Function in context of Worker console
Worker	- executes assigned tasks which are displayed by console. Start, end or reports information on unexpected events, delays, non-conformities to the system or informs the **Team Leader** in person or trough the console
Team Leader	- has same duties as Worker - is responsible for team of Workers - By usage of console he monitors execution of tasks, resolve minor problems and reports to **Station Manager** or **Production/Planning Manager**
System administrator	- is responsible for installing of system modules and giving support to the system health and performance - is not directly involved in production

Table 2: Functions and responsibilities of ARUM roles in context of Worker console

Following Tables 3-14 defines use-cases/requirements which are known and considered within scope of this thesis.

2.3.1 Worker's use-cases

ID	FR-01
Title	Log-in to the application
Role	Worker Team leader
Description	Users allowed to use the application have to log-in via login screen after start of application. The system will load the user information and set permissions according to his role.

Table 3: Functional requirement 1

ID	FR-02
Title	View schedule
Role	Worker Team leader
Description	User will be able to see planned list of jobs dedicated to him by scheduler.

Table 4: Functional requirement 2

ID	FR-03
Title	View job
Role	Worker Team leader
Description	User will be able to see detail information about jobs and their progress.

Table 5: Functional requirement 3

ID	FR-04
Title	Start job
Role	Worker Team leader
Description	User will request system if he may start work on job. When required conditions are fulfilled the system changes state of job and user will be authorised to execution of the job. For example not finished preceding job or missing resource may be unfulfilled conditions.

Table 6: Functional requirement 4

ID	FR-05
Title	Complete job
Role	Worker Team leader
Description	User will be able to report completion of the job.

Table 7: Functional requirement 5

ID	FR-06
Title	Report issue or delay
Role	Worker Team leader
Description	User will be able to report identified issue that he is not able to resolve himself. He will be able to report delay caused by minor issue.

Table 8: Functional requirement 6

ID	FR-07
Title	New schedule notification
Role	Worker Team leader
Description	User will be notified when new schedule is applied. Rescheduling helps in improvement of resources utilization.

Table 9: Functional requirement 7

ID	FR-08
Title	Issue solution notification
Role	Worker Team leader
Description	User will be informed about solution of issue which is present on his dedicated job.

Table 10: Functional requirement 8

ID	FR-09
Title	Issue notification
Role	Team leader
Description	Team leader will be informed about issues reported by workers.

Table 11: Functional requirement 9

ID	FR-10
Title	Displaying of notifications
Role	Worker Team leader
Description	User will be able to display notifications (events) which are in some way important with respect to his role or his work. He will be able to filter them and see details.

Table 12: Functional requirement 10

ID	FR-11
Title	Reaction on issue
Role	Team leader
Description	Team leader will decide whether the reported issue is really issue or not. He will summon meeting of Process managers in former case or allow work execution on associated job in later case.

Table 13: Functional requirement 11

ID	FR -12
Title	Communication with web service
Application	
Description	Application will be able to request data from RESTful web service and persistently store them. Sending data to web service is also required.

Table 14: Functional requirement 12

2.3.2 Further requirements

- Displaying of events. In the context of ARUM and in Worker console, an event is defined as a system notification about failures/nonconformities, their solutions, started and finished jobs, new schedule application etc. All types of events, their categories are listed in appendix C.
- The access to the information through worker console should be enabled after worker's authentication with help of the authentication authority. The security service is a responsible authority that enables the Worker to log-in into the console. The console will be then allowed to communicate with other modules in ARUM.
- The console should be simple, lucid and user friendly.
- Due to expected usage of Worker console in an industrial environment and necessity of portability the targeted devices will be hand-helds.
- Specific for AIB:
  - The Worker reporting the potential nonconformity should start a new job and not wait for a solution.
  - Maximally one job may be executed by worker in one moment. But there may be several halted or terminated jobs dedicated to worker at the same time.
- Specific for IHF:
  - Maximally one job may be running or halted.

- The reason is that RAO/Team Leader decides whether the problem is really non-conformity or not.

2.4 Hand-held consoles

According to previous analysis and requirements in subsection 2.3.2 on developed tool this section will briefly describe the use of consoles in the industrial environment, hardware/software requirements, their properties and present trends.

(a) Pistol-grip handheld [16]

(b) Rugged tablet [17]

(c) Handheld with additional peripheries [18]

Figure 5: Rugged hand-helds

2.4.1 Hardware

Nowadays graphical consoles are widely used in industrial sector. The majority of graphical consoles are embedded and used for monitoring or controlling of manufacturing processes (for example manufacturing automates or robots activities, etc.). The use of mobile consoles in industrial sector grows rapidly. In fact, the use of hand-held consoles has around 20 years of evolution and consoles with different accessories were used for different purposes. Pen equipped PDA computers, Pocket PCs or pistol-grip consoles were pioneers among hand-helds. The practical usage was limited by insufficient connectivity and computational power technology. The devices were expensive because they were designed for single purposes. As an example of widely used single purpose devices are the code readers, which are used by logistics for specification of type, location or amount of goods.

At present, the situation changes with rapid technology progress in personal smart phone market.A great amount of smart phones and tablets emerged on market in the last few years. The mass production of such devices reduced price of the components and it opened a

new direction in industrial hand-helds. Nowadays the trend is to produce universal devices with touch screens (nevertheless in industrial environment hardware keyboard still have their place), customised for rough environment and equipped with wide variety of sensors. Beside special display adaptation, industrial devices differ from personal devices with higher weight, longer battery endurance, waterproof and drop-resistant hardened body. Laser scanners, bar code or RFID readers, wireless radios, industrial interfaces, etc. may be integrated to these devices beside standard commercial equipment.

Following a selection from list of producers providing rugged devices: 2T, Aceeca, Amrel, Bluebird, Casio, Datalogic, Getac, Handheld, Honeywell, Intermec, Janam, Juniper, Motorola, Panasonic, Ruggedbook, Samsung, Trimble, Unitech, Winmate.

2.4.2 Operating system

In the past Microsoft corporation was leader in operating systems for industrial hand-helds. The majority of devices was based on Windows CE or Windows Mobile OS. Later, new large and well supported developer's communities appeared with change in personal cellular phones market. It persuaded producers to support different operating systems like Android or iOS. In presence, number of Android based devices is similar as Windows based devices and trend is that Android slowly replaces Windows in the field of industrial hand-helds.

3 Design

3.1 Preliminary phase

3.1.1 Selected devices and platform

According to the trends in hand-helds it was decided to implement final application on Android platform. For speeding up of the development process it was decided to aim on new devices with Android version higher than 4.0 Ice Cream Sandwich. Devices with display size from 5 inches to 10 inches were decided as target devices. Three different civil devices were chosen for development purposes, but for industrial use rugged alternatives may be considered.

Screen size	Commercial device	Rugged alternative
5"	Samsung Galaxy Note II	Panasonic FZ-X1
7"	Prestigio Multipad	Motorola ET1
10.1"	Lenovo Yoga Tablet 10	Panasonic Toughpad FZ-A1

Table 15: Testing commercial devices and their rugged alternatives

Commercial devices may be suitable for usage in the Iacobucci case, because working places are static desks equipped with a set of tools for several workers. The Workers don't need to carry the devices during operations and no big danger except fall from table threatens the device. The previous statement is not applicable when the device is used by the Team Leader/RAO, because of the requirement on his high mobility.

The situation is different when we consider the Airbus case. The workers move through and around the aeroplane trunk. They have to pick tools and resources before operations and have to carry the device with them. The usage of a rugged device is appropriate in such case.

Other features may be added in the future. For example the the bar-code readers embedded in rugged devices may be used for checking of resource correctness.

3.1.2 Design approach

Based on requirements and required use-cases described in section 2.3, a low fidelity prototype was created. The need of easily changeable sketch of application describing required functionality was the reason for usage of low fidelity prototyping. This helps to define graphical appearance and to design the structure of the application and a user friendly navigation through it. As well, the low fidelity prototype helped to define software components required for creation of functional program and data representing structures.

The tool Balsamiq Mockups [19] was used to create a prototype described in sections 3.2.

3.1.3 Balsamiq Mockups tool

Balsamiq Mockups [19] is a simple graphical tool allowing user to design the interface of websites, desktop or mobile applications. The whole low-fidelity model consists of basic objects frequently used in the user interfaces. It means objects like buttons, selectors, text views, sliders, and many others. The advantage was that Balsamiq allows the designer to export the whole model to an interactive PDF, where the mentioned basic objects on one page may be hyper linked with the other pages. Using this kind of navigation through the PDF, the real use-cases may be effectively presented to the end users.

3.2 User interface design

The dimensions of the screen and target device were not specified at the time of first prototype creation. However, it was decided that the target device's operational system will be Android. Therefore the prototype fitting to classical cell phone (4" screen) was created. The reason was that since Google released Android 3.0 feature of fragments was introduced. The use of fragments allows the programmer to use same code and graphical design in applications running on devices with different screen sizes. The details about use of the fragments will be described in section 4.3.3

The pink highlighted items on screen-shots in this section may be noticed. The pink highlights have no purpose from view of final application, but they are used in the low-fidelity prototype to let user know that the coloured field is interactive.

3.2.1 Login screen

The figure 6 shows the initial screen, which appears after the start of the application. See requirement in table 3. On the initial screen the user verifies his/her identity with the help of a security service. The security service also provides information about user's authorization. The information about user's identity will be temporally saved in application's internal memory.

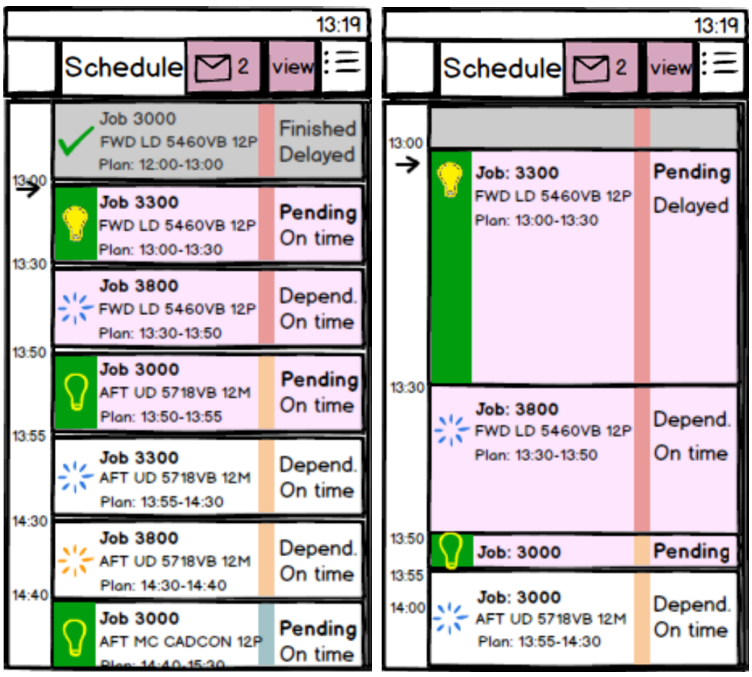
Figure 6: Login screen

3.2.2 Lists of jobs

The lists of jobs are are shown in the figures 7a,7b. With respect to requirement defined in table 4 the worker is able to see the list of jobs assigned to him. It was decided to create two different perspectives to make the application user friendly. The main reason was that the durations of the particular jobs are different.

The perspective shown in figure 7a shows an uniform size of items in the list. This is practical when the worker wants to see how many jobs he has been assigned to and in what state the jobs are.

The perspective shown in figure 7b helps the worker to asses how long separate jobs are. The original intention was to use a linear scale function to emphasise, that the job with the double list item size has a double duration compared to another job. Nevertheless it would lead to unpleasant results when some jobs may be even 500 times longer then others. Therefore the parametric scale was used.



(a) Uniform view (b) Parametric view

Figure 7: List of jobs

Within the list of jobs the user may see general information such as the state of job, the time state, the durations and the character of job. The values and the types of the showed information are defined and described in section 3.4.

In addition to the list of jobs the screens contain time axis showing actual time. It should help the worker to monitor how much time left he has to complete the job with respect to his schedule.

The click on item from the list leads to new screen with the job detail. See subsection 3.2.4.

3.2.3 Action bar

The action bar with the action buttons is visible in figure 8. The action bar itself is dedicated for navigation trough the screens and it also contents buttons of often used actions.

On the very left there is a navigational button to previous screen. In the home screen of the application (List of jobs) this button is empty.

On the second place from the left a name of screen is shown. It helps to simplify the orientation in the application.

The button with the envelope and a number shows the quantity of the events that may affect the user's activity. See requirements in tables 9-11. With a click on the envelope icon a new screen with Event list opens. See subsection 3.2.5. The events are defined in section 2.3.2.

The click on button *View* changes perspective of the job list.

The last button shows a drop down list with the other possible action buttons that are not so often used. Among them are the screen settings and the application settings.

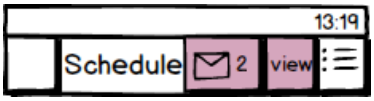


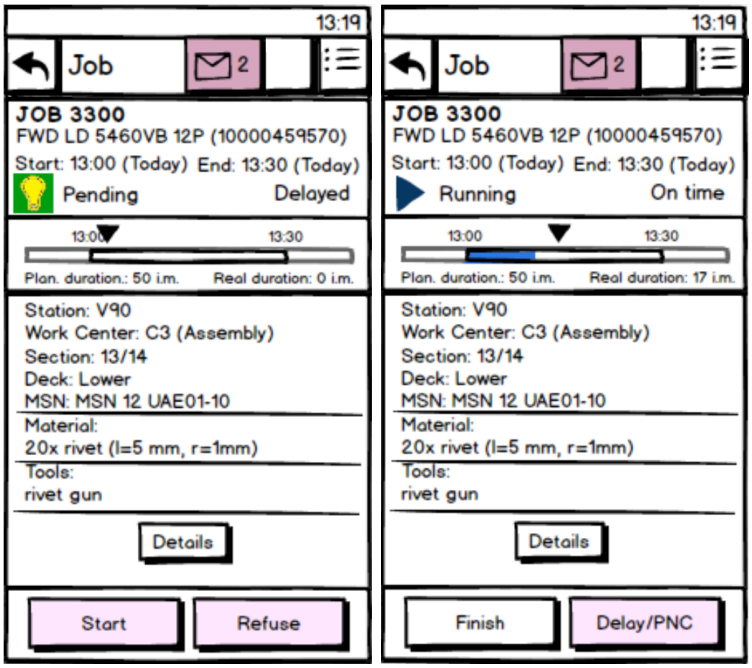
Figure 8: Action bar

The example of the navigation with use of the action bar is displayed in section 3.2.8.

3.2.4 Job detail

Figures 9a,9b display information about one particular job dedicated to the worker. According to requirement in table 5. Both screenshots show one job, but each of them in different time and with different state. Below the action bar the general overview of the job info is present on top segment of screen.

The time bar showing the planned duration of the job, the actual time and the real duration of the job is placed below the general overview.

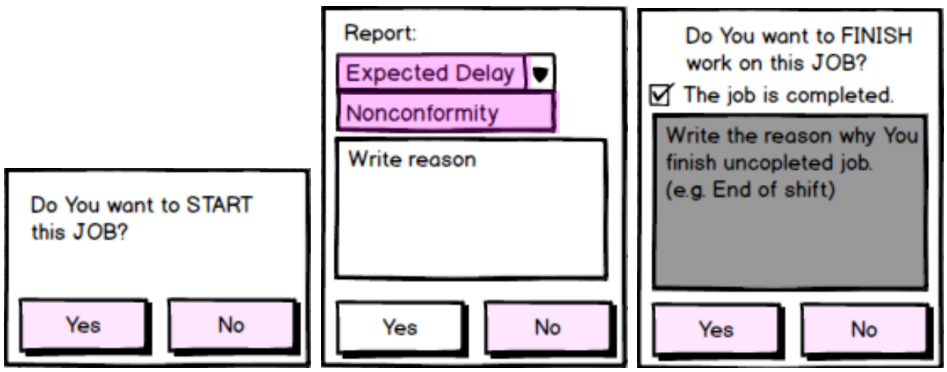


(a) Pending job (b) Running job

Figure 9: Particular job screens

The detailed information about job is written in center of the screen. The specification of the place of work, the required resources as a material and the tools are listed. Other co-workers participating on the particular job, special instructions and notes may be also displayed. Blueprints or technical documentations may be displayed by pressing of the "details" button if available.

The buttons on the bottom of the screens define operations on the job. The use of these buttons fulfils the requirements defined in tables 6-8. The transition between possible job's states are described in section 3.4. With a click on this buttons the dialogs verifying chosen action appears. See figures 10a-10c. The dialogs may collect information required for fulfilment of chosen action.



(a) Starting dialog (b) Reporting dialog (c) Finishing dialog

Figure 10: Dialogs verifying chosen action

3.2.5 List of events

The events relevant for ARUM project and Worker console are defined in section 2.3.2. Figure 11 shows the design of events list. The events may affect working routine of the Worker when it reports for instance new schedule or solution of nonconformity. See requirements in tables 9-10. The team Leader may be informed about potential nonconformities and he may react on them. This use-cases of Team Leader fulfil requirements defined in tables 11-13.

Figure 11 shows details about event type, time or corresponding job in case that event is related to one. In case of this mock-up the click on item shows new screen with detail of Event described in section 3.2.6. In addition the functionality of events filtering was considered. Click on button with icon of filter opens screen with advanced settings of filters. The filter screen design may be found in section 3.2.7.

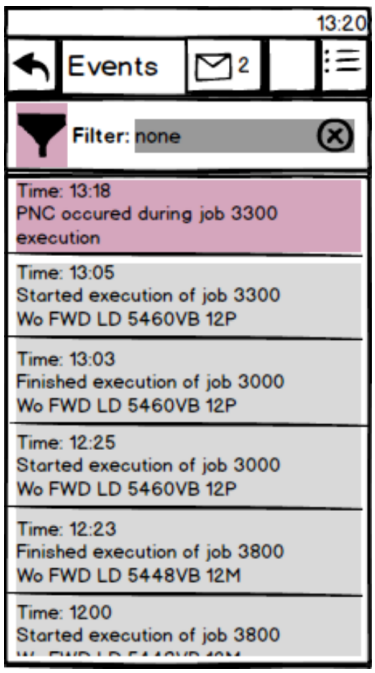
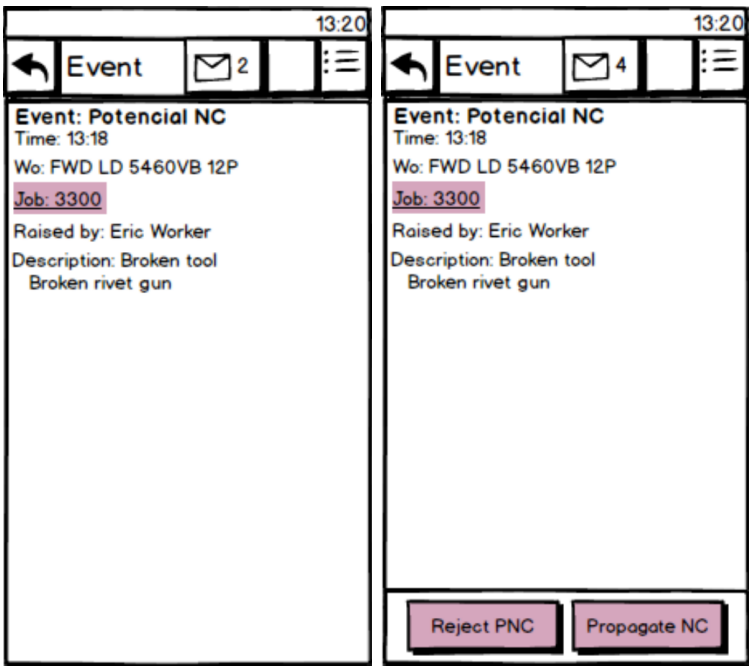


Figure 11: List of events

3.2.6 Event detail

There are visible screens of event detail in figure 12. The screens display the same event, but the left one is displayed on console authorised by Worker and the right one is displayed on console authorised as Team Leader. On both, there are visible details as event type, time, person who raised event, corresponding job and description of the event. The job field is clickable and leads to detail of the job. See subsection 3.2.4. The Team Leader's screen enables the Team Leader to propagate a nonconformity or reject a potential one. That fulfils requirements described in tables 12,13.



(a) Worker console (b) Team leader console

Figure 12: Detail of events

3.2.7 Filter settings

The screen 13 shows custom settings of filter for filtering of the events. A basic functionality of the filtering was considered during for the scope of this thesis. Further enhancement by a more complex filtering schemes is a potential for future development of Worker console. In current state there is a possibility to filter events by type, time of creation or importance of the event. This screen enables the user to create a new customised filter and select, apply or delete the existing one.

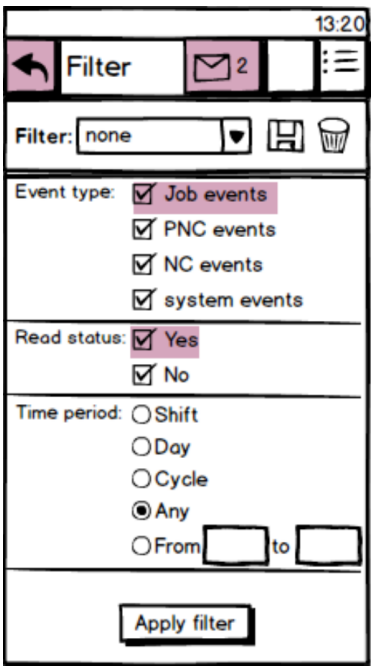


Figure 13: Filter screen

3.2.8 Navigation through screens

The figure 14 shows a diagram of navigation trough the screens of application. Navigation is also illustrated in hyperlinked mock-ups. The mock-ups are visible in appendices D,E at the end of this thesis and also in the enclosed CD/DVD. It was decided to illustrate navigation in its own figure 14, because hyperlinks do not work when the PDF file is printed and the CD/DVD may not be reachable when thesis is in electronic form.

During the development the need of complex navigation was identified as the hiearchy structure is not sufficient. Into the real application a shortcut navigation button was implemented to reach the home screen (Job list) on one click. This functionality will be essential because of frequent usage by the Team Leader.

Figure 14 shows application with the jobs in a constant state. It may happen that new event comes or the user changes state of one job. That scenarios are described only in mock-ups in appendices D,E or in the enclosed CD/DVD.

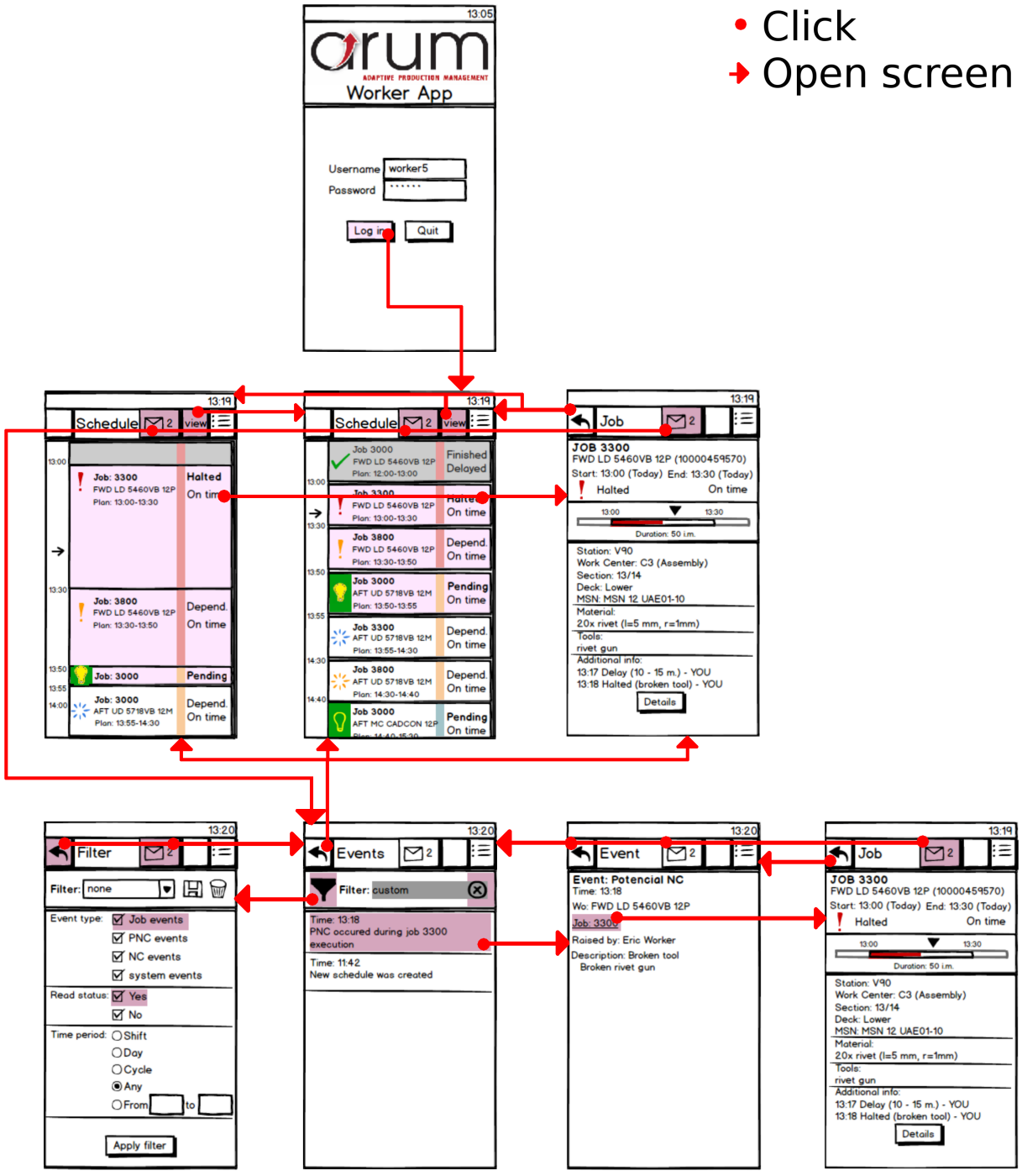


Figure 14: Navigation trough screens

3.3 Design of application architecture

The use of MVC pattern for Android application development is supported by structure of Android project itself, as may found in references [20], [21].

The integration of developed Worker console to ARUM architecture defines communication with Client web service. The web service was designed according to REST architecture, which is beneficial for distributed systems. Therefore the necessity of communication layer within architecture of Working console become obvious. The brief description of Client service is in section 2.1.5.

It was decided that the console will be a thick client. The reason is that hardware requirement to the Client service server, which will be the data provider to hundreds of consoles, will be lesser with use of thick clients. The decision corresponds with the reality, that hand-held devices have usually overdimensioned computation power and no additional costs will be necessary. However the thick clients requires a persistent data storing. The Android platform allows a persistent data storing only by a local database or by serialization into a file in the device memory. Therefore the usage of relational database was proposed.

Proposed architecture of the developed tool is shown in figure 15.

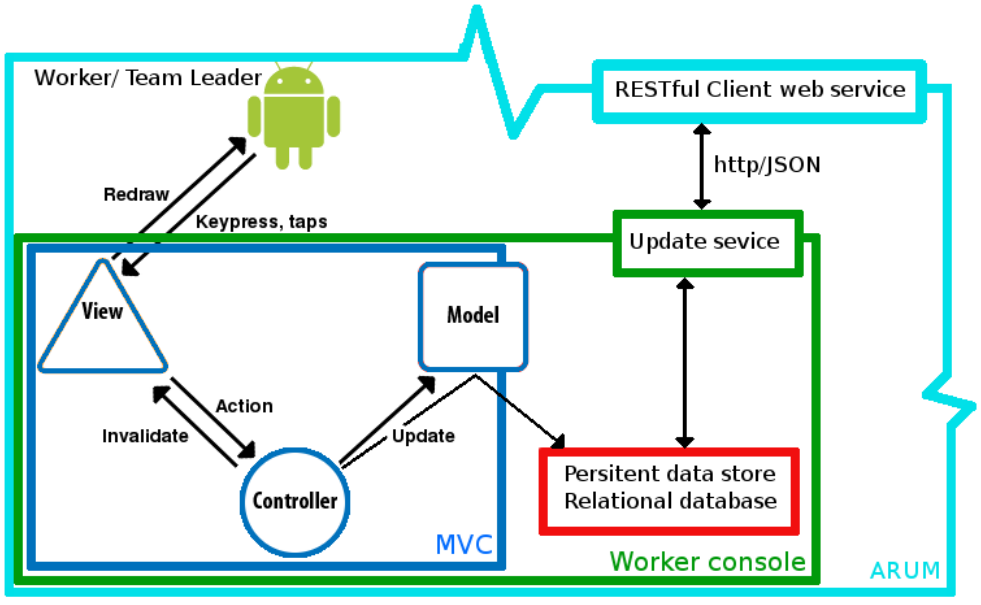


Figure 15: Proposed architecture of Worker console

3.4 Data representation

The analysed requirements and created mock-ups of user interface helped us to design a structure of data representation, that will be used in real application. The proposed model

is based on RESTful web Client service which will be the gateway between the console and ARUM system. The proposed model is simplified enough to meet requirements of our user interface.

3.4.1 SQL database

The entity-relation diagram 16 shows a structure of a relational database which is proposed to be a main data store in the targeted hand-held device.

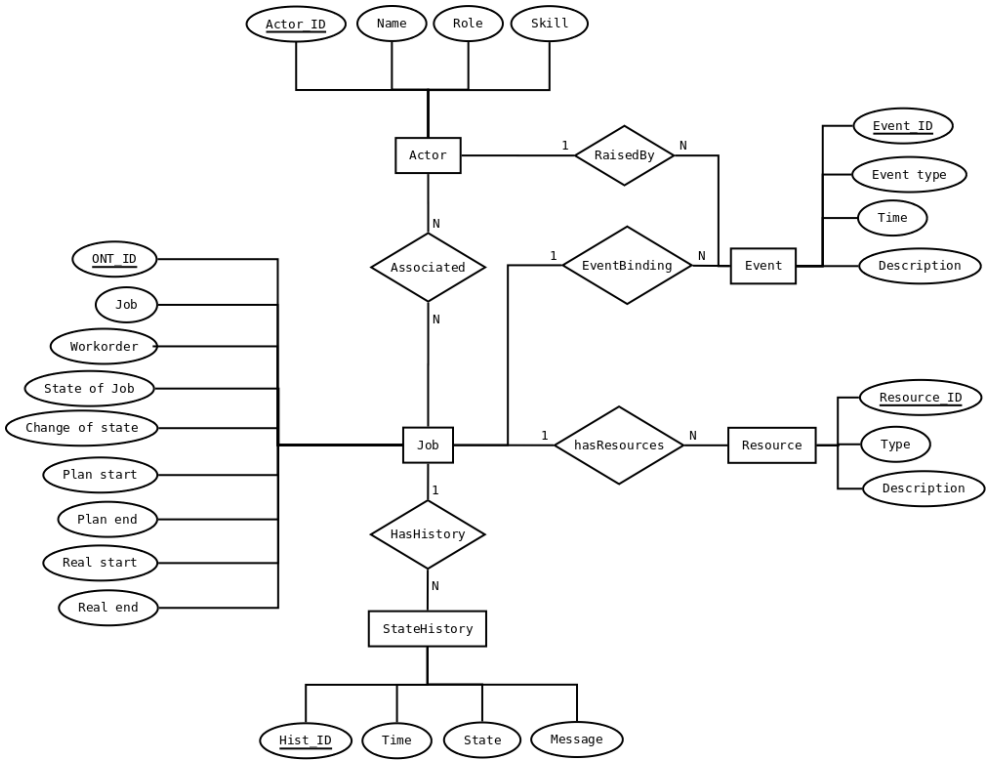


Figure 16: ER diagram

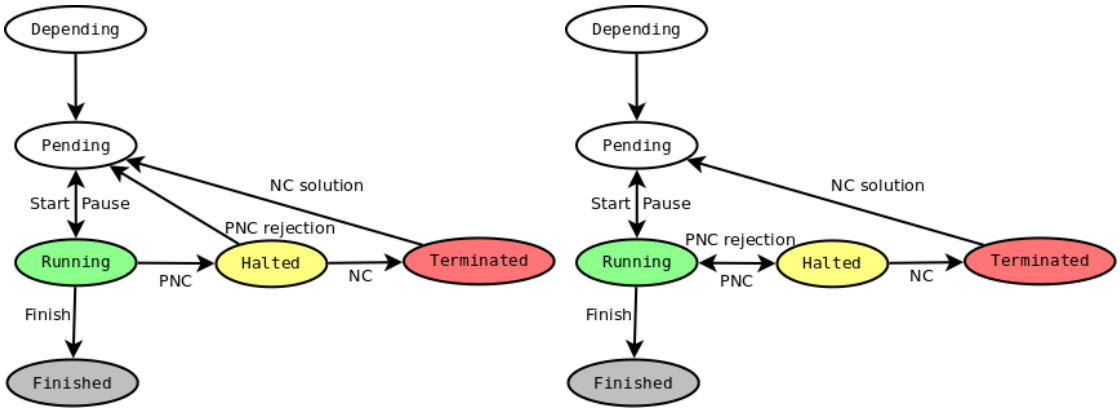
3.4.2 States of job

The possible job states were projected with respect to model Client service (REST) and are visible in table 16. The state depending used in Worker console is added into the set of states defined by Client service model.

States					
Depending	Pending	Running	Finished	Halted	Terminated

Table 16: Possible states of job

Transitions between possible states of job in each use cases is visible in figures 17a,17b. The difference between the two diagrams is caused by different routines of dealing of nonconformities inside AIB and IHF. See section 2.3.2.



(a) AIB state evolution (b) IHF state evolution

Figure 17: Diagrams of state evolution

3.4.3 Time states of jobs

Time states of jobs are defined by table 17. This time state should help the Worker with the orientation if he is on time or delayed in comparison with his schedule.

State	Time state	
	On time	Delayed
Running	$Time_{real} < Plan_{end}$	$Time_{real} \geq Plan_{end}$
Finished	$Real_{end} < Plan_{end}$	$Real_{end} \geq Plan_{end}$
Depending Pending Halted Terminated	$Time_{real} < Plan_{start}$	$Time_{real} \geq Plan_{start}$

Table 17: Possible time states of job

3.4.4 Resource types

Resources are entities that may be required for processing of task. Types of resources were taken from Client service/REST model. The set of resource types contains Facilities, Machines, Materials, Parts, Products and Tools. In the Client service/REST model the Actor is also type of resource but in mobile application model it was decided to make a special object describing the Actor. The reason is that the object describing Actor is the essential for login into the Working console and for gathering of associated jobs.

3.4.5 Event types

The events relevant for ARUM project and the Worker console are defined in 2.3.2 and they are listed in appendix C. Events in ARUM project hold informations about changes in whole system. There was selected a set of events that will be usable for Worker/Team Leader within the mobile application.

The selected set includes the events informing about changes of job states, events holding details about possible nonconformities, nonconformities and their solutions. Beside mentioned ones there were also selected events informing about new schedules and resource shortages.

4 Implementation

4.1 Development platform

The Android is an open-source computational platform that apart from Android operation system includes a wide development support in form of development tools and drivers specification. The whole project is led by Google corporation and supervised by Open Handset Alliance consortium. The operation system itself is written in C, C++, Java languages and is based on the Linux architecture.

Applications for the Android operating system may be developed in Java Programming language with the use of Android SDK or in C/C++ with the use of Native Development Kit. It was decided to write final application in Java language.

4.2 Android Software Development Kit

The Android SDK is a cross-platform development tool providing API libraries and developer tools required for building, debugging and testing of new Android applications written in Java.

The applications may be developed in any text editor and then with use of command line tools (Java Development Kit, Apache Ant and Android Debug Bridge) the Android application APK may be created and installed to mobile device. However The SDK is officially supported by Eclipse integrated development environment and even other IDEs support programming of Android applications.

IDEs provide debugging, graphical tools, editors, that helps developing.

4.3 Worker console project

The project of the Worker console was organised with respect to application architecture design proposed in section 3.3.

Widely used Model-View-Controller (MVC) pattern was amended by another two layers ensuring persistent store of data and communication with Client web service.

The class and files in each layer are briefly described below. For further details about implementation look upon the project, which is enclosed on CD/DVD.

4.3.1 View layer

The design of the application (View layer) is defined in xml files which may be found in folders `/res/layout` and `/res/menu` in our project.

```
--/res
  |--/layout
  |--/menu
```

The implementation of layout was based upon agreed design of user interface described in section 3.2 and shown in appendices D,E. Other subfolders in folder `/res` contain pictures and files with strings that may be used in definition of screens layouts or lately in program as changing entities.

The advantage of text strings separation from Java code become useful when application is used in different countries. File `Strings.xml` in folder `/res/values` represents default (English) set of strings. However there may be placed another file with German set of strings `Strings.xml` into folder `/res/values-de`. The application selects the appropriate set of text strings with respect to the used device language.

4.3.2 Model layer

The layer contains Java classes which define the data structure. This classes may be found in folder (packages):

```
--/src/com/arum
  |--/model
```

Variables of all classes from the package `com.arum.model` are illustrated in figure 16.

4.3.3 Controller layer

All Java classes of the controller layer are placed in folders

```
--/src/com/arum
  |--/activities
  |--/fragments
```

Classes within both folders are quite similar. Beside selection of layout which will be shown on screen (selection of View), both kinds of classes implement listeners which are activated by user activity on the screen. The major differences between Activities and Fragments is that the former one may stay alone, but the later one may not. It means that Activity is a kind of container which may contain zero, one or multiple Fragments. This functionality allows the developer to design one application for use on the devices with different size of screens.

The example of Activities with one Fragment are visible in figures 18a, 18b. The Activities with two Fragments may be seen in figures 19,20. The Activities with no fragment may look exactly the same as in case of one Fragment in Activity, however the programmer is not allowed to use designed layout in another Activity.

4.3.4 Database layer

The Java classes from database layer may be found in folder (packages):

```
--/src/com/arum
  |--/databases
```

The classes in package `com.arum.common` handle local SQLite database which is used as a main storage within the mobile device. The structure of the SQLite database is illustrated in figure 16.

4.3.5 Update service layer

The Java classes from Update service layer may be found in folder (packages):

```
--/src/com/arum
  |--/service
```

The class UpdateService from `com.arum.service` package implements background service which handles communication with the Client web service trough http requests. The data are transferred between server and client in JSON format. The UpdateService class uses prepared library which handles connection and communication with web Client service and transforms the responses received from the server into Java objects and vice versa. Author of mentioned library is Ondřej Harcuba, who is another active participant in the ARUM project.

5 Testing

The usability testing procedure, it's aims and results are described in this section. The details of the testing may be found in appendix B.

5.1 Description of test

The implemented application was tested by six volunteers and covered several simple working routines of Worker/Team leader in IHF company. The testing of application by one tester took about eighty minutes. The testers were familiarized with the working routines in Iacobucci before the start of the testing.

For each testing the internal time of device was changed and the state of application was arranged to the default state for each testing scenario. That ensured repeatability of testing and the test results were not influenced by different appearance and state of application.

5.1.1 Inconsistencies in testing

It is necessary to mention that the testing was not completely independent and few recommendations about testing were violated.

The supervisor of testing was the same person, who developed the application. The effort was made to make the testing unprejudiced and to discover as many inconsistencies and missing functionalities as possible.

Another inconsistency during the testing was, that most of the testers were previously introduced to the IHF production processes and some of them are active in development of another components in ARUM. Nevertheless the remarks recorded by an independent tester were similar to the remarks of the others.

5.2 Process of testing

The complaints, remarks, misunderstandings and wrongly executed operations were recorded during all testing. The testing composed of following phases.

5.2.1 Describing phase

The application and device was set up to default state for testing scenario 1.

The different screens of application were presented to testers. They were asked to describe, what they see on the screens, what control elements they identify and what they expect to happen after activation of mentioned elements.

5.2.2 Testing scenario 1

The application and device was set up to default state for testing scenario 1.

The tester was asked to do the simple operations, which are regularly repeated in the role of Worker in IHF company. The operations were finishing of the actual running job, starting following job and reporting of PNC.

The illustration of whole testing scenario 1 is in figure 21 in section 6

5.2.3 Testing scenario 2

The application and device was set up to default state for testing scenario 2. The state of application differs from the end of the first scenario only in change of job state (from Halted to Running). That simulated the rejection of PNC by the Team leader. The tester was asked to identify how and why the state of application changed from the end of scenario 1.

The illustration of whole testing scenario 2 is in figure 22 in section 6

5.2.4 Testing scenario 3

The application and device was set up to default state for testing scenario 3. The tester was asked to use the application as the Team leader. The tester was to identify the state of application and to find out the incoming PNC reported by his inferior. After that he was asked to reject the PNC and propose the reason of PNC rejection.

The testing scenario 3 is also illustrated in figure 23 in section 6

5.3 Results

All testers executed the tests successfully and without any problem.

After the analysis of remarks from the testing (see documents on CD/DVD) the list of requests on the changes of application was created. The list is visible in appendix B.

There were found number of requests on change of graphics and text formulations, that could improve lucidity and user friendliness.

Some functional requirements, that are missing in the application or are implemented in a misleading manner, were discovered. Among mentioned functional requirements were:

- Showing of jobs on which the focused job is dependent and their job states.
- Displaying events associated to jobs directly and vice versa.

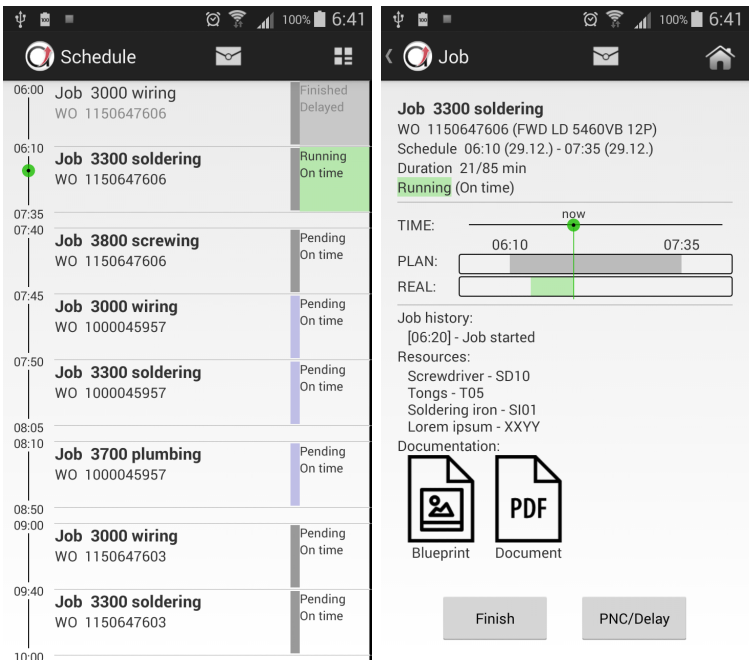
- Insufficient functionality of implemented filter of events.
- More vehement notification of incoming events, that may be important for the user.
- Possible hiding of events generated by the user himself.
- Simplification of un/marking of un/important events.
- Logout from the application after the end of the shift.

Some objects that lead to misunderstandings during the first contact with application were identified. The purpose of these objects were clarified during the usage and the objects were considered as not important. Most frequent was the mismatch between times of planned jobs and their real execution. Mentioned issues are not so bothersome and they even might be solved by application tutorial.

6 Demonstration

The demonstration of the real functionality of the application is presented in this chapter. However the best practice how to get know the application is testing on own device. The installation guide is enclosed on CD/DVD and in appendix F.

The figures 18 shows portrait oriented screens of the application, where the one shows associated schedule and the second shows detail of a running job.



(a) List of jobs (b) Detail of job

Figure 18: Application screens in portrait orientation

The figures 19,20 shows portrait oriented screens. The first screen shows schedule associated to worker, the second one shows the same schedule and detail of the selected job in the second part of the screen.

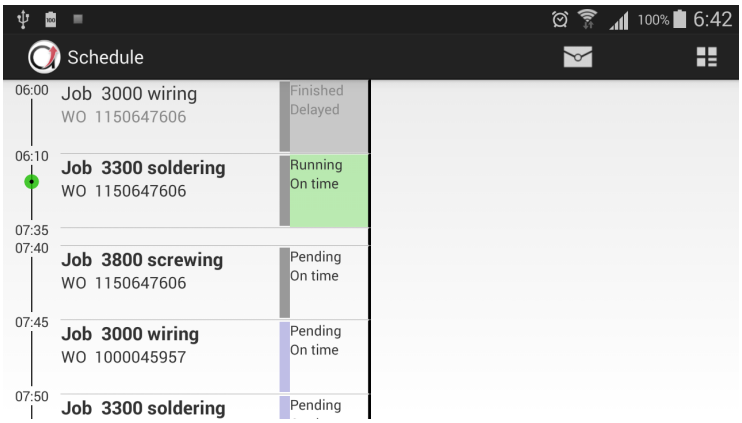


Figure 19: Screen with schedule in landscape orientation

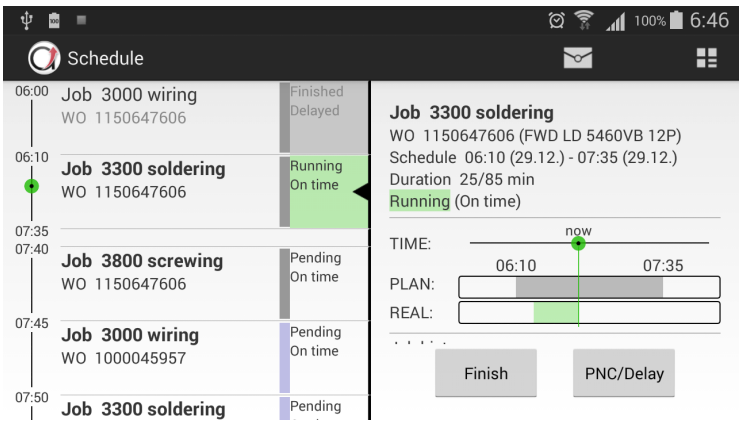


Figure 20: Screen containing schedule and job detail

The diagrams illustrating testing scenarios 1-3 and also covers almost full functionality of the application. See figures 21-23.

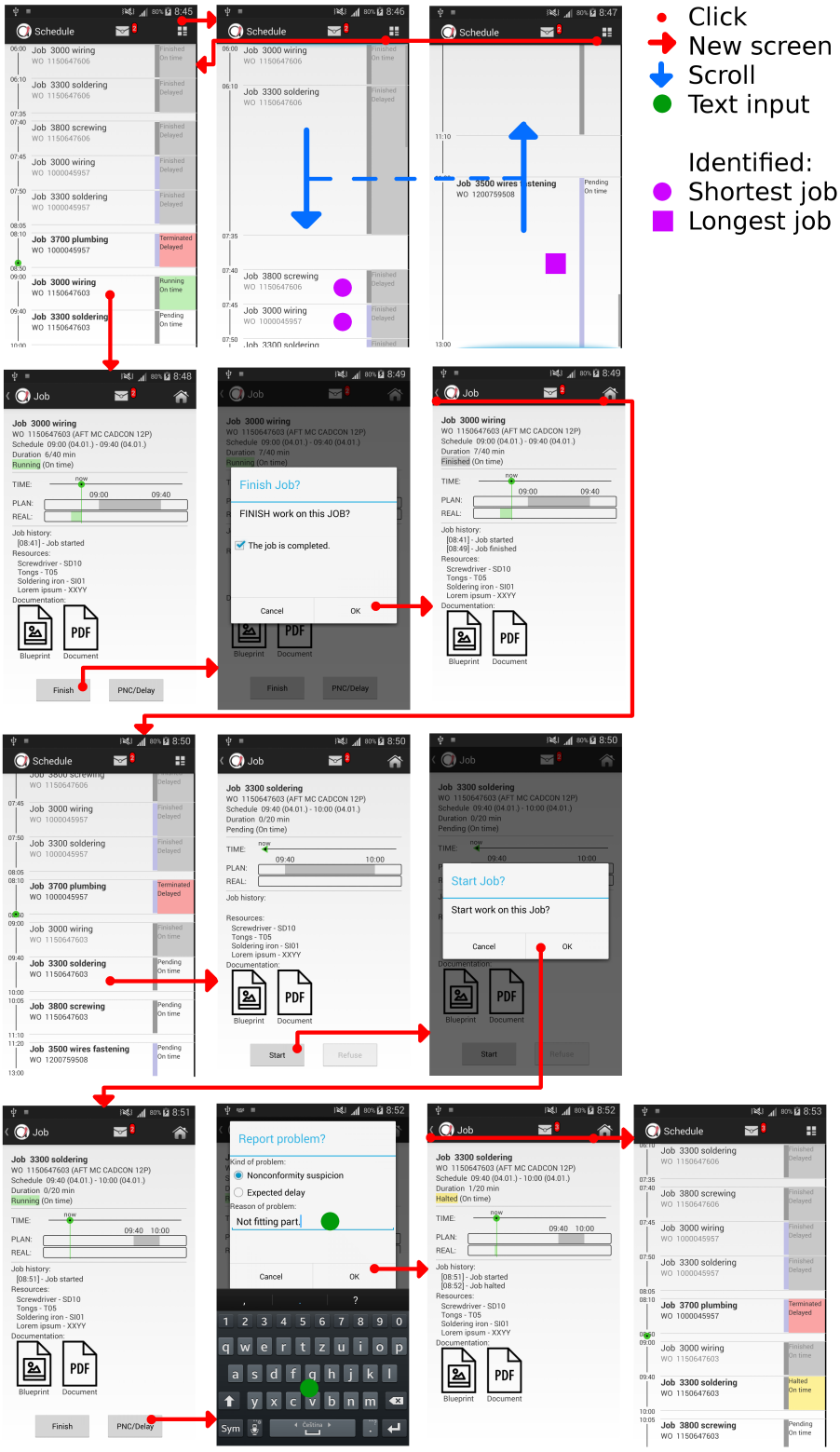


Figure 21: Testing scenario 1 (IHF, role Worker)

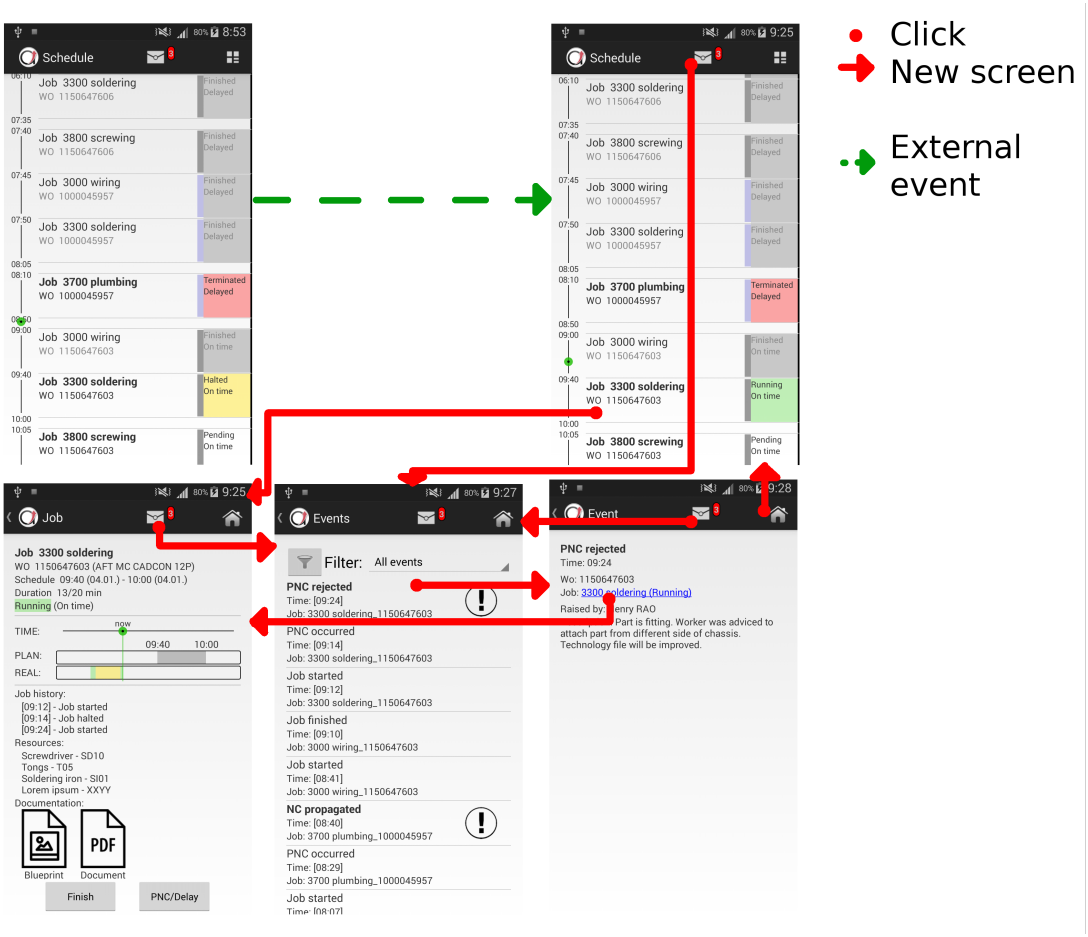


Figure 22: Testing scenario 2 (IHF, role Worker)

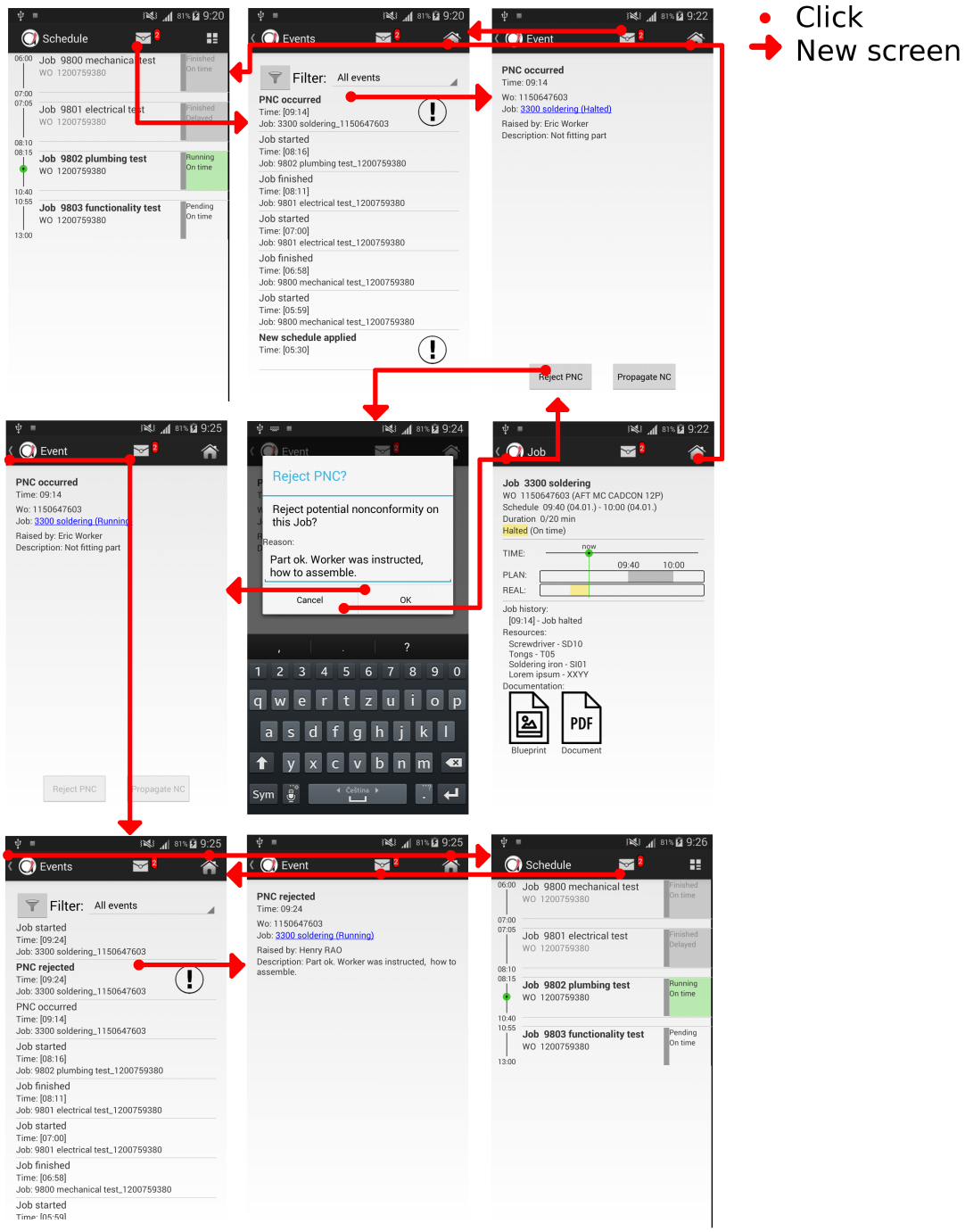


Figure 23: Testing scenario 3 (IHF, role Team Leader)

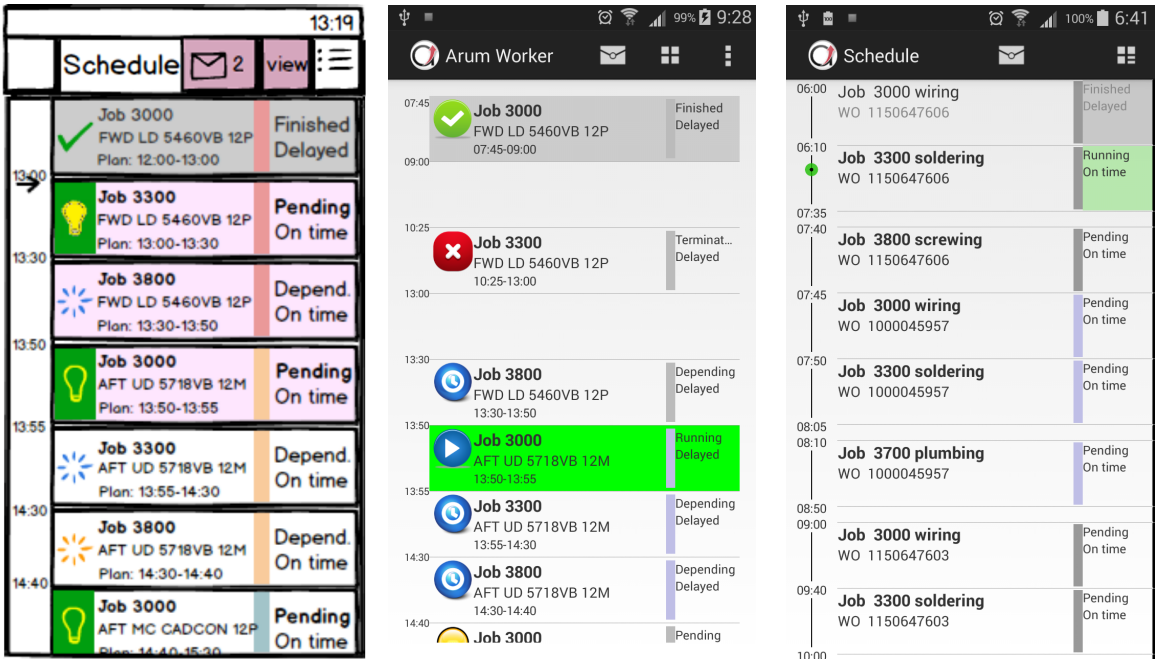
7 Conclusion

7.1 Thesis summary

The aim of this thesis was to design, implement and test a tool that will be used for the assembly line worker's activity management in Airbus and Iacobucci companies and will enhance the ARUM project solution.

The development of the required tool was performed in several phases, including analysis, design phase with prototyping, implementation and testing partially displayed in figures 24a-24c.

As a result a tool called Worker console was developed. The Worker console includes all specifications and requirements, which were possible to be integrated at the current stage of development of ARUM project (January 2015). Only the login of the user into company network trough Worker console is not included as the Security service is under external development. As ARUM is still in development the Worker console cannot be fully integrated into the overall project architecture.



(a) Mock-up (b) First implementation (c) Actual implementation

Figure 24: Evolution of application

7.2 Future work

For full integration and handover of the Worker console to the end user further functionalities have to be implemented.

The additional requirements from the end users show the necessity to implement the above mentioned security functionality, to implement displaying and completing of alternative tasks to the Worker (AIB) and to add a functionality of the final quality check (IHF). This development The usability testing discovered necessity of various changes in graphical design improving lucidity and user friendliness. Also changes in functionality which are mainly connected to displaying of events and their associations with jobs must be added. The development work on the Worker console will continue troughout 2015.

References

[1] Europian Commision. Seventh framework programme (fp7). `http://cordis.europa.eu/fp7/home.html`, 2014. [rev. 2014-11-4].

[2] Europian Commision. Factories of the future. `http://goo.gl/coa5gM`, 2014. [rev. 2014-11-4].

[3] Sevent Framework Programme. Grant agreement for: Collaborative project arum, 2012. [Grant agreement no:314056].

[4] EADS. Arum project. `http://arum-project.eu/`, 2012. [rev. 2014-11-5].

[5] ALEXANDER BIELE. Use-case definition (use-cases 1 & 2) - arum d2.1.1, 2013. [confidential].

[6] EADS. System architecture and platform specifications - arum d4.1.1, 2014. [confidential].

[7] Marin C.A., Monch L. , Leitao P. , Vrba P. , Kazanskaia D. , Chepegin V. , Liwei Liu , Mehandjiev N. A conceptual architecture based on intelligent services for manufacturing support systems. In *Systems, Man, and Cybernetics (SMC)*, number 14067503 in IEEE International Conference, pages 4749 – 4754, Manchester, 2013. IEEE.

[8] Pavel Vrba, Petr Kadera, Martin Myslik, Martin Klima. Jboss esb sniffer-message flow visualization for enterprise service bus. In *Industrial Electronics (ISIE)*, number 14483050 in Proceedings of International Symposium on Industrial Electronics, pages 1724 – 1729, Istambul, 2014. IEEE.

[9] Luis Moreno. Tool for production processes operational monitoring. Bachelor's thesis, Czech Technical University in Prague, 2014.

[10] Michal Fuksa. Methods and tools for inteligent esb. Master's thesis, Czech Technical University in Prague, 2014.

[11] David A Chappell. *Enterprise service bus*. O'Reilly Media, Inc., 2004.

[12] Roy Thomas Fielding. *Architectural Styles and the Design of Network-based Software Architectures*. PhD thesis, University of California, Irvine, 2000.

[13] Ondřej Harcuba. Tool for operational monitoring of production processes. Master's thesis, Czech Technical University in Prague, 2014.

[14] Giovanni E. Pazienza Jos de Jong, Ludo Stellingwerff. Eve: a novel open-source web-based agent platform. In *Systems, Man, and Cybernetics (SMC)*, number 14067650 in IEEE International Conference, pages 1537 – 1541, Manchester, 2013. IEEE.

[15] AIRBUS. Airbus orders and deliveries summary 2014. http://www.airbus.com/presscentre/corporate-information/orders-deliveries/, 2014. [rev. 2014-11-5].

[16] Janam company. www.janam.com. [photo],[rev. 2014-11-5].

[17] Panasonic company. www.panasonic.com. [photo],[rev. 2014-11-5].

[18] Juniper company. www.junipersys.com. [photo],[rev. 2014-11-5].

[19] Balsamiq. http://balsamiq.com/products/mockups/. [rev. 2014-11-17].

[20] Borek Bernard. http://www.zdrojak.cz/clanky/uvod-do-architektury-mvc/. [rev. 2014-12-27].

[21] Ben Jakuben. http://teamtreehouse.com/library/build-a-blog-reader-android-app/exploring-the-masterdetail-template/the-modelviewcontroller-mvc-design-pattern-2. [rev. 2014-12-27].

Glossary

AIB	-	Airbus
ARUM	-	Adaptive Production Management (Name of the European Union project)
EU	-	European Union
ESB	-	Enterprise Service Bus
FNSD	-	Factory Network Scenario Designer
ICT	-	Information and communications technology
IT	-	Information technology
IHF	-	Iacobucci
KPI	-	Key Performance Indicator
MAS	-	Multi-agent systems
MR	-	Missing Resource events
NC	-	Nonconformity
NC suspicion	-	Potential nonconformity
PNC	-	Potential nonconformity
RAO	-	Operative Area Responsible
Ramp-up period	-	Phase of transition between product development and maximum capacity utilization.
REST	-	Representational State Transfer
SAP	-	Enterprise Resource Planning system
Task	-	Generic class of Job or Work order
UI	-	User Interface

Usability testing of ARUM Worker Console
DEFINITION

Description

The usability testing is focused on simulation of simple procedures within IHF product assembly routine.
This is the first testing of application and is targeted to people who have no or minimal experiences with mentioned application.
The aim of this testing is to find how user-friendly the application is for not experienced user.
Secondary aim is gathering of tester's opinions of possible improvements and missing functionalities.

Introduction to assembly procedure

Roles

There are two roles which are important in scope of this testing.
The first role is **Worker**. The Worker does the jobs which are associated with him by scheduler. List of job is provided by application and Worker reports start/end of job with use of application. He also may call responsible worker (Team leader) in case of problem (**PNC**) during his work.

The second role is **Team leader**/(RAO). The Team leader is Worker with extended competence and responsibility. He does jobs assigned to him and he is called when some problem during Worker's job execution occurs. He consider if the problem really is problem (**propagate NC**) or not (**reject PNC**).

Events

Change of states, report of problem or other actions generates events which holds information what actually happened. Team leader is notified about problems with help of events. Worker is notified when the problem is resolved or rejected.

Evolution of job states

There are figured possible states of jobs on Figure 25.

- When worker **starts** job in application, state of job changes from **Pending to Running**.

- When he **finishes** job in application, state of job changes from **Running to Finished**.
- When he **reports PNC** in application, state changes from **Running to Halted** and responsible Team leader is called.
- Team leader may **reject PNC** and bring job back to **Running** state or he may **propagate NC** and bring job to **Terminated** state.
- Job may be brought from state **Terminated** to **Pending** by person with role Process manager. This case is out of scope of this testing.

Only one job may be **Running** or **Halted** in the same time.

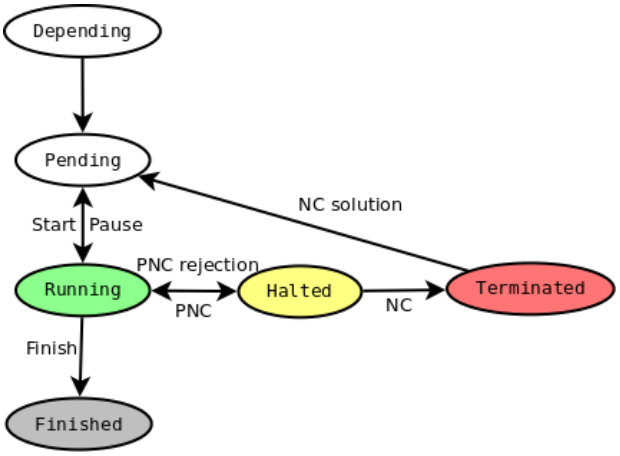


Figure 25: Navigation trough states of job

Glossary

PNC - potential nonconformity - nonconformity suspicion
NC - nonconformity
RAO - Team leader

Introduction to role

Tester is Worker/(Team Leader) in Iacubucci company and works from 6:00 to 13:00. Now is 9:00 and he already done some work. Now he returns from brake and launch the application.

Testing

Preliminary questions

- Are you familiar with use of smart hand held devices (Phones, tablets)?
- Are you familiar with Android operating system? If not, what OS do you use on your devices?
- Do you have some experiences with development of mobile device applications?

Start of application

Start application with icon showed on figure 26.

Figure 26: Launcher

Description of screens

Without any knowledge about application the tester should describe presented screens, identify control elements and describe what the elements do when they are activated. Application is set to scenario 1 by supervisor and clocks of device are set to time 8:45.

- Schedule
- Second completed job
- Running job
- Eventlist
- Event

Exploration of application

Explore application on your own.

Setting by supervisor 1

Application is set to scenario 1 by supervisor and clocks of device are set to time 8:45. The tester was informed that his role is Worker.

Testing scenario 1

-Identification of shortest and longest job
-Start of next possible job
-Reporting of PNC with message *Not fitting part.*

Setting by supervisor 2

Application is set to scenario 2 by supervisor and clocks of device are set to time 9:25.

Testing scenario 2

-Define what happened from last time.

Setting by supervisor 3

Application is set to scenario 3 by supervisor and clocks of device are set to time 9:20. The tester was informed that his role changed to role Team Leader.

Testing scenario 3

-Define what happened during your work.
-Reject PNC. You informed Worker how to do the task correctly.

Questions

-Evaluate how friendly and lucid the application is for you.
-Evaluate navigation trough the application.
-What was least lucid?
-What didn't you understand most?
-What functionality did you miss most?

Results summary

Preliminary questions

Familiar with smart devices - 6/6
Familiar with Android - 6/6
Experiences with development - 1/6

Requests on text changes

-SCHEDULE - 2 - What difference is between Terminated and Finished?(1 - same color for both)
-JOBDETAIL - 2 - Do not approve second same string "Job started" in history. Proposed "unhalted/restarted".
-JOBDETAIL - Button string "Finish" change to "End job"
-EVENTLIST - 2 - Does not approve unique identifier of job (1 doesn't want to read),(1 do not understand meaning)
-EVENTDETAIL - PNC rejection - "rejected by" instead of "raised by".
-DIALOGS - 2 - redundancy in dialog - message, title
-DIALOG finishing - Possibilities "DONE-FINISHED"

Requests on graphical changes

-SCHEDULE - alternative look icon
-JOBDETAIL -plan has the same color as finished job
-JOBDETAIL - 3 - fonts of subtitles. (Resources, history - all melts together)
-JOBDETAIL - expect big change after change of job state
-JOBDETAIL -Add icon declaring state of job.
JOBDETAIL - Proposed to place durations inside progress bars.
-EVENTLIST - 2 - not lucid. (1 Propose changing color of rows)
-EVENTLIST - important - Exclamation on left side
-EVENT ICON - 2 - Not understand envelope. (1 Propose Text "EVENTS" or bell icon)
-EVENTS - Add symbol declaring type of event. -EVENT - clickable link - opening expecting in web browser. Proposed button.
-EVENT - overlooked clickable link
-DIALOGS - capital letters, different styles of texts in dialog

Requests on functionality changes

-JOBDETAIL - Missing quantity and units in list of resources.
-JOBDETAIL - 2 - Missing Dependant jobs.

-JOBDETAIL - 3 - direct link from job to corresponding events.
-JOBDETAIL - 2 - time - day only in case that day is different from present one.
-JOBDETAIL - separation of days in progress bar
-JOBDETAIL - 2 - planned duration mark in real progress bar
-JOBDETAIL - expected start/end time of job drew in progress bar.
-JOBDETAIL - expected explanation why start of next job was not possible.
-DIALOG - 2 - PNC dialog may be confirmed with empty reason. (1 - Expected next dialog confirming selected choice and written text).
-EVENTS - 2 - Missing filtering events by job.
-EVENTS - 5 - sound/vibrating signalisation of new event.
-EVENTS - 2 - Signalisation of new event during minimised application.
-EVENTS - 5 - Pointless displaying of own events (1 proposed incoming, outgoing)
-EVENTS - 4 - auto-deactivation of important flag after click on important event (possibility of activation)
-EVENT - 3 - Missing history of events/associated events
-EVENT - 2 - JOBDETAIL - RAO FUNCTIONALITY - missing name of worker, description of problem, buttons in job detail.
-APPLICATION - 2 - Who is logged in. Who I am.
-APPLICATION - 3 - Log out from application/ end of shift. (1 - propose end of unfinished jobs after notification) (1 - dialog after click on back button from Schedule)
-APPLICATION - 2 - Required summarising screen KPI of worker and KPI of product.

Misunderstandings

-SCHEDULE - 3 workorders
-SCHEDULE - 5 color wo grouping.(1 propose stronger divider in list)
-SCHEDULE - 4 -**Confused plan and real execution of jobs**- Nearly everytime found meaning
-SCHEDULE - what is meaning of "delayed"?
-SCHEDULE - delay wrongly connected with plan duration, not plan end.
-SCHEDULE - 3 - confusing alternative view icon
-SCHEDULE - 2 - alternative view - (height of displayed jobs shows how important job is)(some summary of work)
-SCHEDULE - what screen label Schedule means? is it clickable?
-SCHEDULE - 2 - Expected more information after click on grey finished field, or red terminated field.
-BACK ICON - Contain text. Doesn't know what to expect.
-JOBDETAIL - 2 - Confused real/plan duration (1 propose labels)(1 propose only planned duration)
-EVENTS - Are all displayed events associated to worker?
-FILTER - What does time in filter set up?
-FILTER - How will be filter created and applied?

Other requests and uncertainties

-SCHEDULE - alternative display of jobs Schedule /real execution
-SCHEDULE - why is one Terminated -expected note "problem in progress"
-SCHEDULE - expect actualisation of list when list is dragged down (email)
-JOBDETAIL - After finishing of job expected automatic return into schedule screen.
-Eventdetail - Would like to see time of expected solution of problem.
-PNC dialog - Possible to select kind of PNC from list.
-EVENTS ICON - Red background associated with terminated job.
-EVENT - After click on Jobdetail he expected highlighted moment in progress bar.
-APPLICATION -Propose side hiding panel with navigation and settings elements. (email application)
-RAO - Did not know if Team leader should report delay on his job/ stop his job.

List of events in ARUM

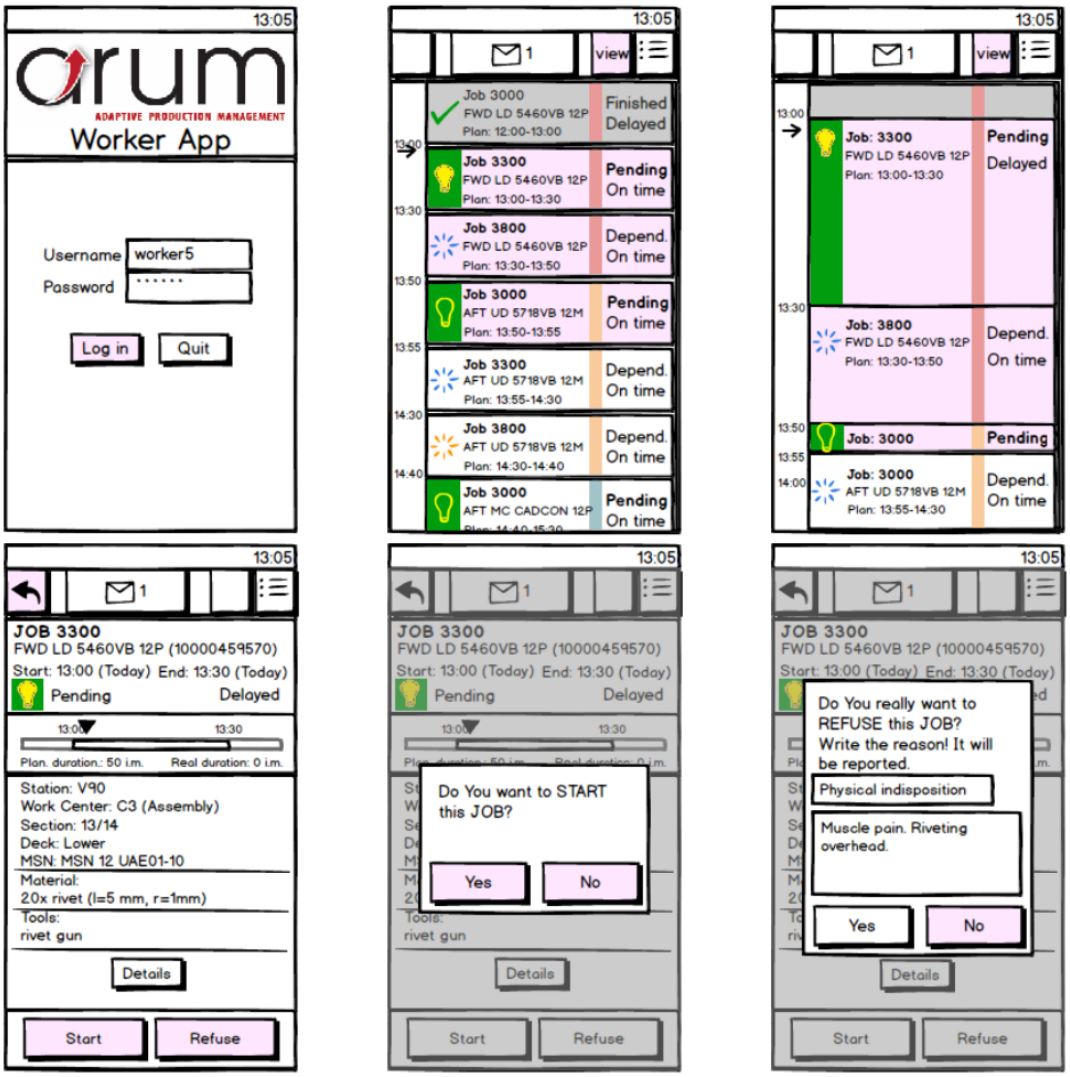
addedResourceEvent
decreasedInventoryEvent
defaultCycleTimeUpdatedEvent
finishedJobEvent
haltedJobEvent
increasedInventoryEvent
jobEvent
jobPlannedEndTimeChangedEvent
jobPlannedStartTimeChangedEvent
jobPropertyChangeEvent
liveSceneCycleUpdatedEvent
missingResourceEvent
missingResourceTypeEvent
ncMeetingDecidedEvent
ncMeetingEndedEvent
ncMeetingScheduledEvent
ncMeetingStartedEvent
ncMeetingUpdatedEvent
ncRefinementEvent
ncRelatedEvent
ncSolutionEvent
ncSuspicionEvent
ncSuspicionRejectionEvent
newAvailabilityEvent
newJobEvent
newScheduleAppliedEvent
newScheduleAvailableEvent
newSkillEvent
newWorkorderEvent
nonConformityEvent
pmAcceptedEvent
pmRejectedEvent
pmRequestedEvent
raoAcceptedEvent
raoCheckReworkEvent
raoCheckSuccesfulEvent
raoRejectedEvent
raoRequestedEvent
removedAvailabilityEvent
removedResourceEvent
removedSkillEvent

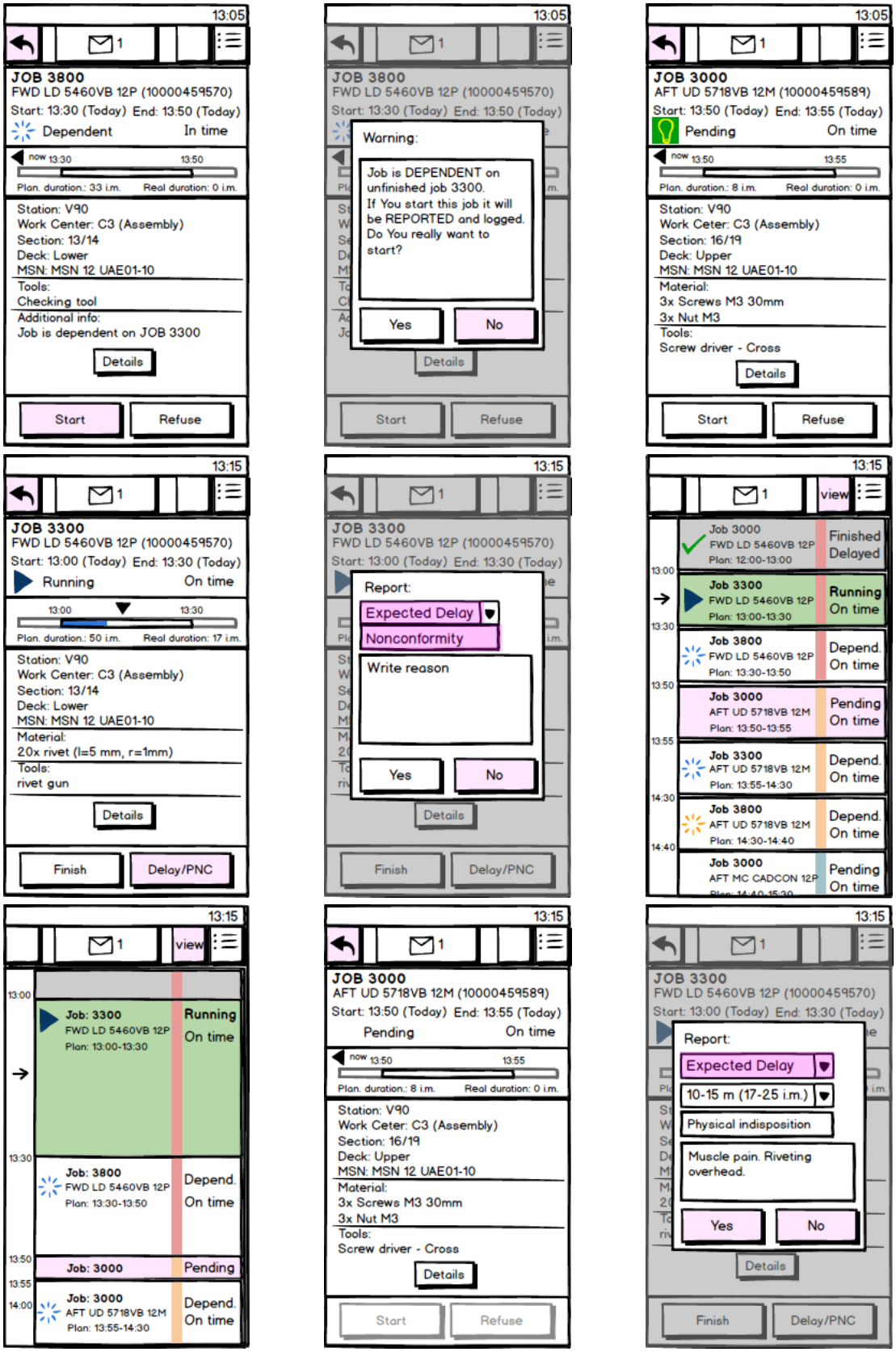
removedWorkorderEvent
resourceEvent
resourceTypeEvent
runningJobEvent
sceneEvent
terminatedJobEvent
transferredResourceEvent
updatedAvailabilityEvent
workorderEvent

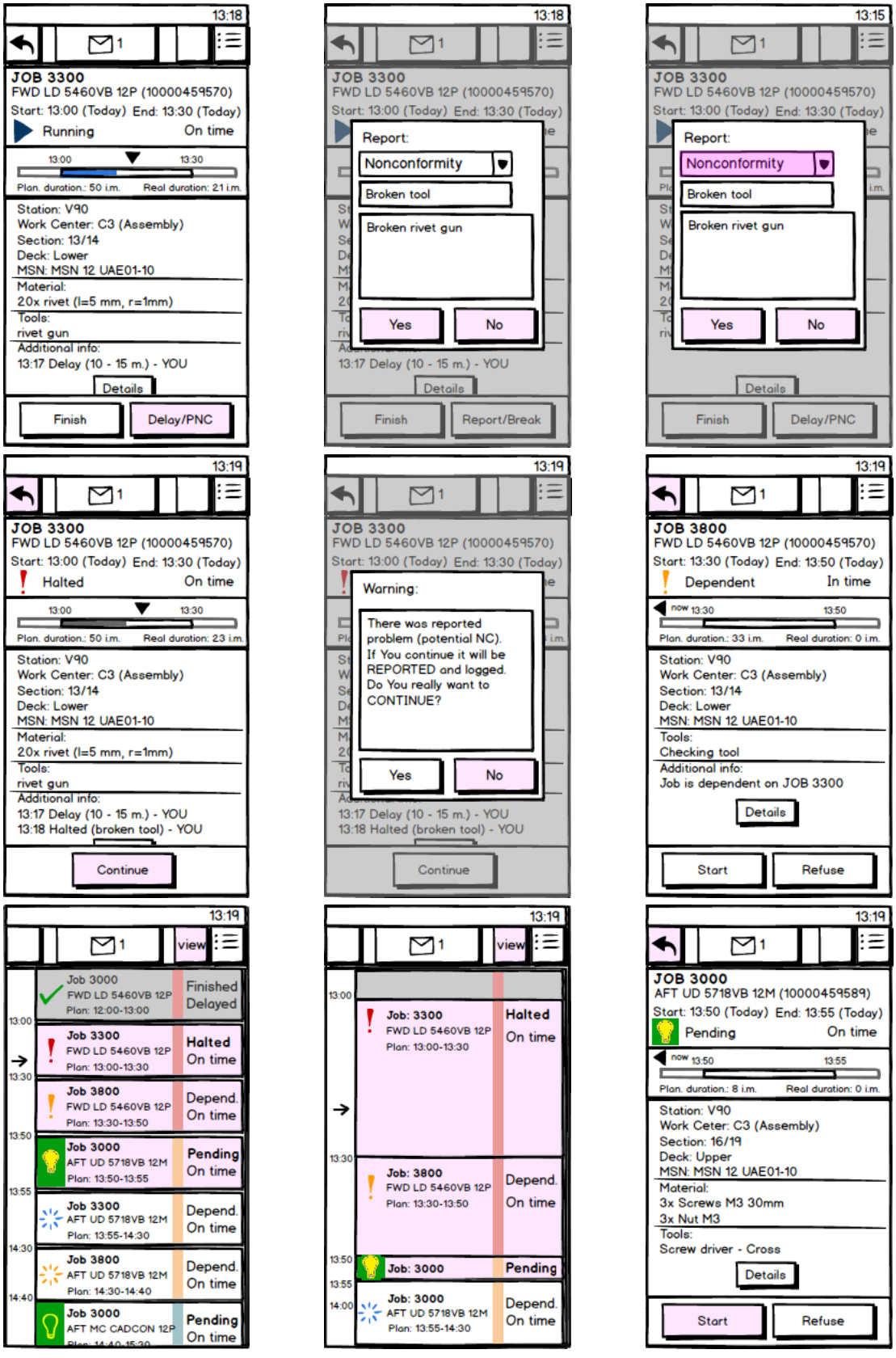
Type of the Event

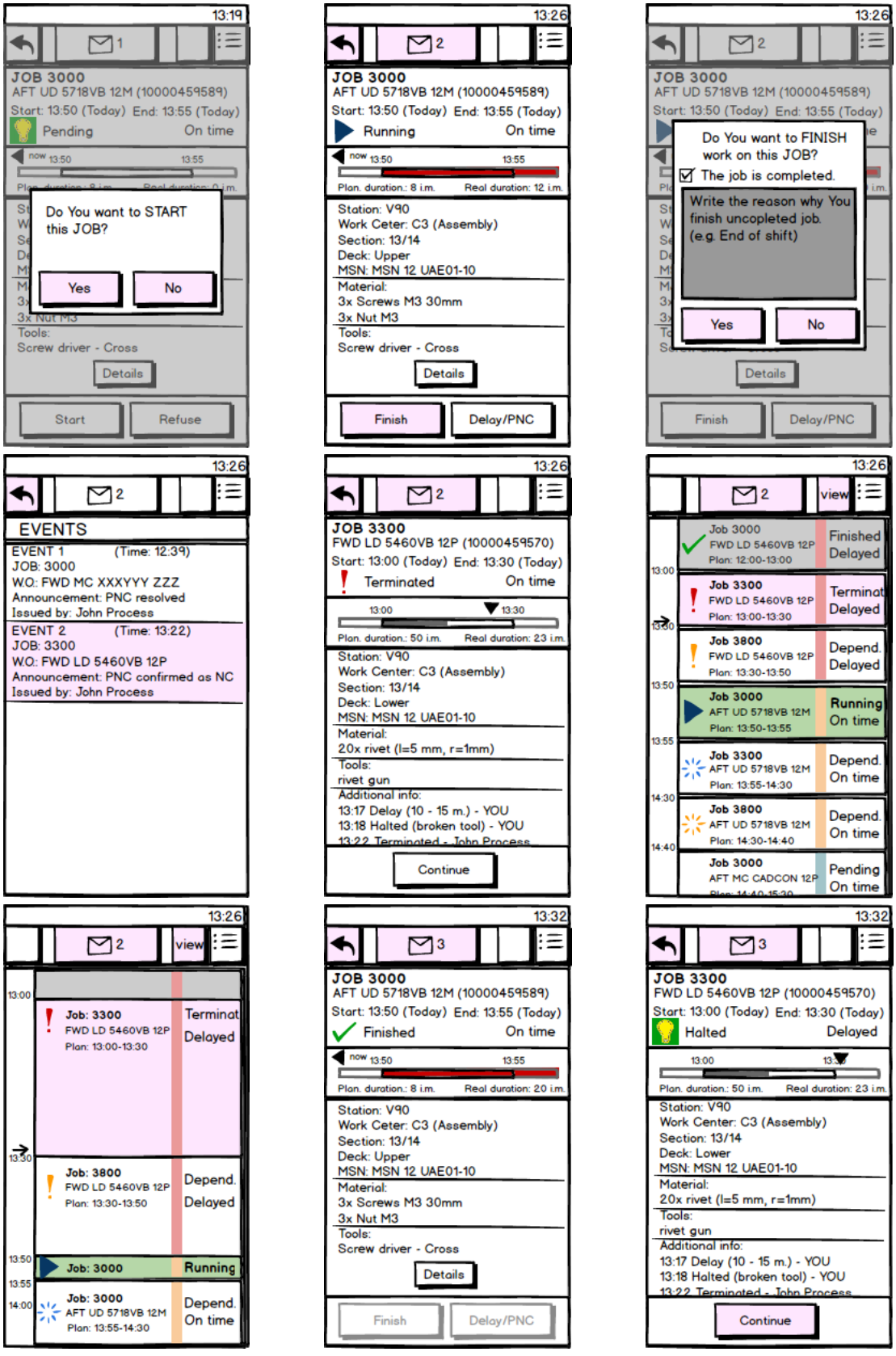
JOB_EVENT
NC_EVENT
PM_EVENT
RAO_EVENT
RESOURCE_EVENT
RESOURCE_TYPE_EVENT
SCENE_EVENT
WORKORDER_EVENT
EVENT

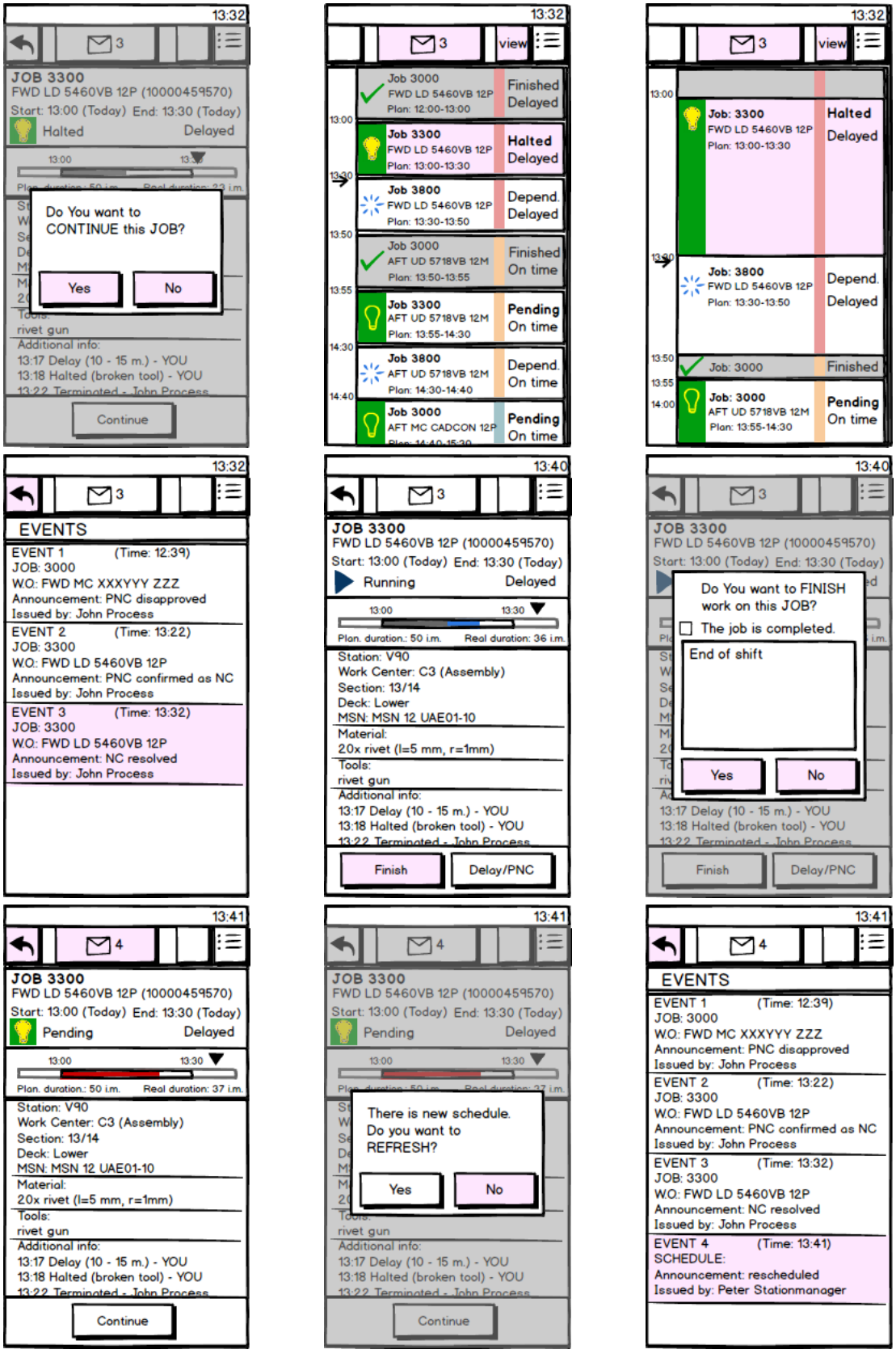
Mock-ups Worker

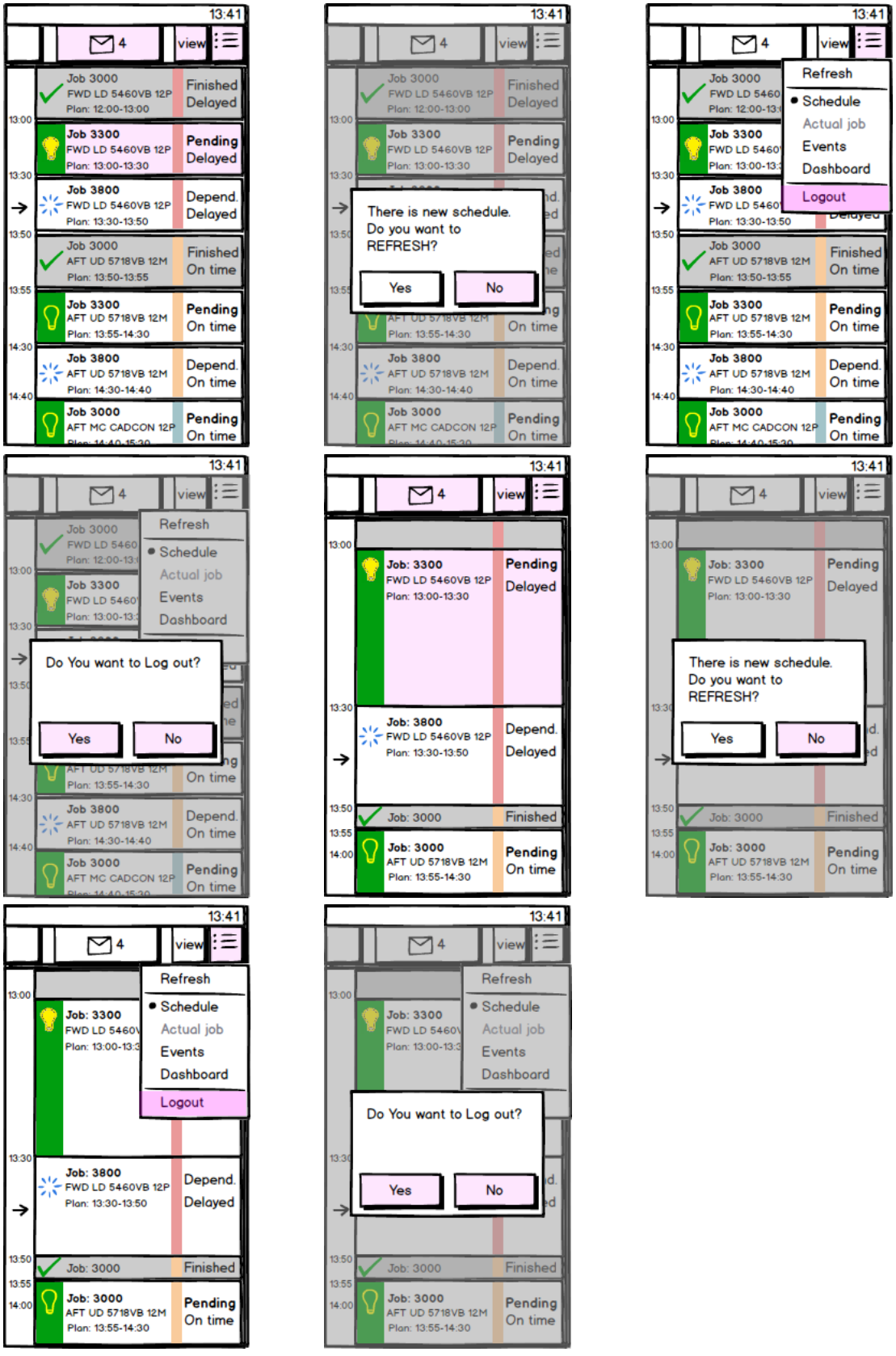



13:05
JOB 3800
FWD LD 5460VB 12P (10000459570)
Start: 13:30 (Today) End: 13:50 (Today)
Dependent
In time
now 13:30
13:50
Plan. duration.: 33 i.m.
Real duration: 0 i.m.
Station: V90
Work Center: C3 (Assembly)
Section: 13/14
Deck: Lower
MSN: MSN 12 UAE01-10
Tools:
Checking tool
Additional info:
Job is dependent on JOB 3300
Details
Start
Refuse
Warning:
Job is DEPENDENT on unfinished job 3300.
If You start this job it will be REPORTED and logged.
Do You really want to start?
Yes
No
JOB 3000
AFT UD 5718VB 12M (10000459589)
Start: 13:50 (Today) End: 13:55 (Today)
Pending
On time
now 13:50
13:55
Plan. duration.: 8 i.m.
Real duration: 0 i.m.
Station: V90
Work Ceter: C3 (Assembly)
Section: 16/19
Deck: Upper
MSN: MSN 12 UAE01-10
Material:
3x Screws M3 30mm
3x Nut M3
Tools:
Screw driver - Cross
13:15
JOB 3300
FWD LD 5460VB 12P (10000459570)
Start: 13:00 (Today) End: 13:30 (Today)
Running
On time
13:00
13:30
Plan. duration.: 50 i.m.
Real duration: 17 i.m.
Station: V90
Work Center: C3 (Assembly)
Section: 13/14
Deck: Lower
MSN: MSN 12 UAE01-10
Material:
20x rivet (l=5 mm, r=1mm)
Tools:
rivet gun
Finish
Delay/PNC
Report:
Expected Delay
Nonconformity
Write reason
view
Job 3000
FWD LD 5460VB 12P
Plan: 12:00-13:00
Finished
Delayed
Job 3300
FWD LD 5460VB 12P
Plan: 13:00-13:30
Running
On time
Job 3800
FWD LD 5460VB 12P
Plan: 13:30-13:50
Depend.
On time
Job 3000
AFT UD 5718VB 12M
Plan: 13:50-13:55
Pending
On time
Job 3300
AFT UD 5718VB 12M
Plan: 13:55-14:30
Depend.
On time
Job 3800
AFT UD 5718VB 12M
Plan: 14:30-14:40
Depend.
On time
Job 3000
AFT MC CADCON 12P
Pending
On time
Job: 3300
FWD LD 5460VB 12P
Plan: 13:00-13:30
Running
On time
Job: 3800
FWD LD 5460VB 12P
Plan: 13:30-13:50
Depend.
On time
Job: 3000
Pending
Job: 3000
AFT UD 5718VB 12M
Plan: 13:55-14:30
Depend.
On time
Expected Delay
10-15 m (17-25 i.m.)
Physical indisposition
Muscle pain. Riveting overhead.

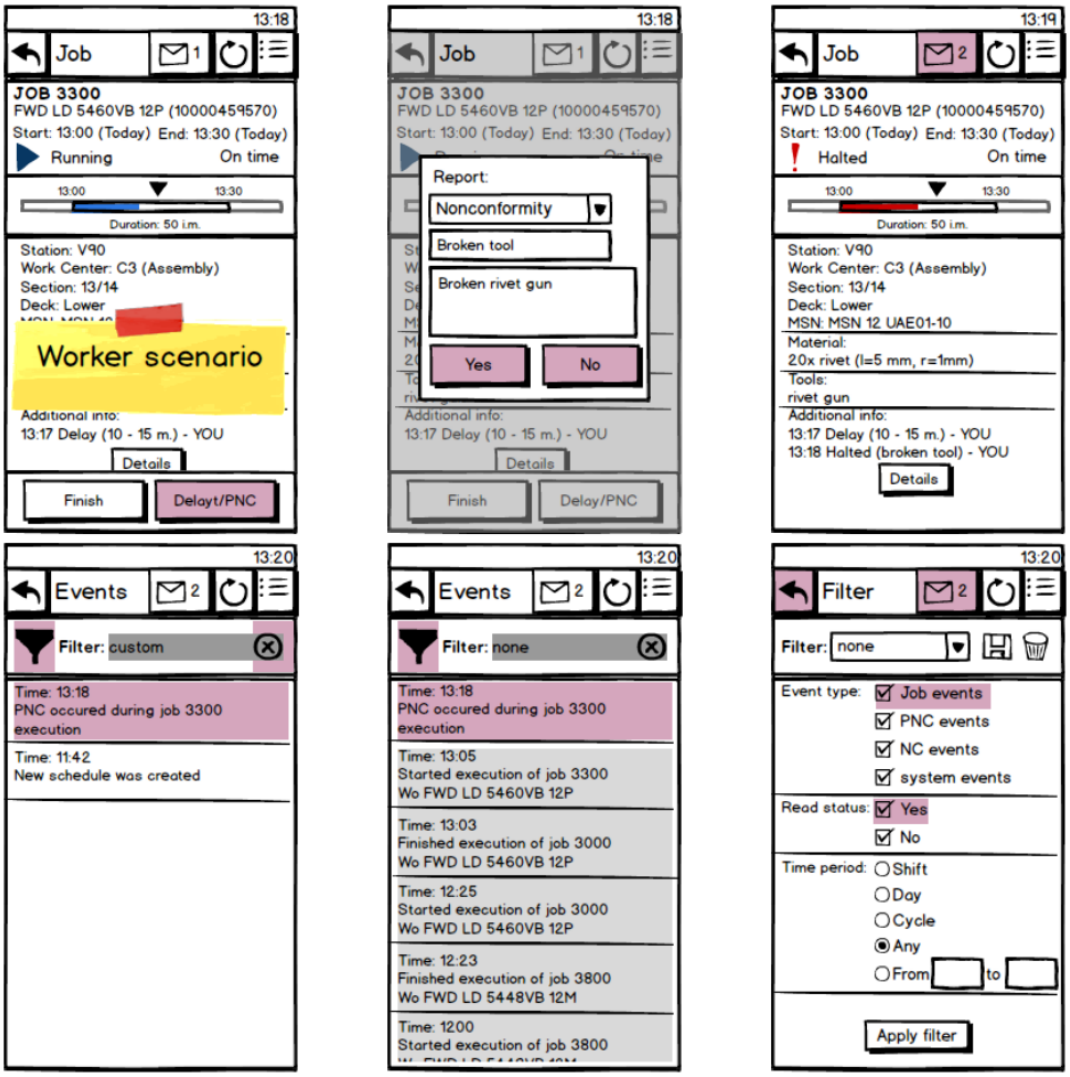

13:18
JOB 3300
FWD LD 5460VB 12P (10000459570)
Start: 13:00 (Today) End: 13:30 (Today)
Running
On time
13:00
13:30
Plan. duration.: 50 i.m.
Real duration: 21 i.m.
Station: V90
Work Center: C3 (Assembly)
Section: 13/14
Deck: Lower
MSN: MSN 12 UAE01-10
Material:
20x rivet (l=5 mm, r=1mm)
Tools:
rivet gun
Additional info:
13:17 Delay (10 - 15 m.) - YOU
Details
Finish
Delay/PNC
13:18
Report:
Nonconformity
Broken tool
Broken rivet gun
Yes
No
13:17 Delay (10 - 15 m.) - YOU
Details
Finish
Report/Break
13:15
Report:
Nonconformity
Broken tool
Broken rivet gun
Yes
No
Details
Finish
Delay/PNC
13:19
JOB 3300
FWD LD 5460VB 12P (10000459570)
Start: 13:00 (Today) End: 13:30 (Today)
Halted
On time
13:00
13:30
Plan. duration.: 50 i.m.
Real duration: 23 i.m.
Station: V90
Work Center: C3 (Assembly)
Section: 13/14
Deck: Lower
MSN: MSN 12 UAE01-10
Material:
20x rivet (l=5 mm, r=1mm)
Tools:
rivet gun
Additional info:
13:17 Delay (10 - 15 m.) - YOU
13:18 Halted (broken tool) - YOU
Continue
13:19
Warning:
There was reported problem (potential NC). If You continue it will be REPORTED and logged. Do You really want to CONTINUE?
Yes
No
13:17 Delay (10 - 15 m.) - YOU
13:18 Halted (broken tool) - YOU
Continue
13:19
JOB 3800
FWD LD 5460VB 12P (10000459570)
Start: 13:30 (Today) End: 13:50 (Today)
Dependent
In time
now 13:30
13:50
Plan. duration.: 33 i.m.
Real duration: 0 i.m.
Station: V90
Work Center: C3 (Assembly)
Section: 13/14
Deck: Lower
MSN: MSN 12 UAE01-10
Tools:
Checking tool
Additional info:
Job is dependent on JOB 3300
Details
Start
Refuse
13:19
view
13:00
13:30
13:50
13:55
14:30
14:40
Job 3000
FWD LD 5460VB 12P
Plan: 12:00-13:00
Finished
Delayed
Job 3300
FWD LD 5460VB 12P
Plan: 13:00-13:30
Halted
On time
Job 3800
FWD LD 5460VB 12P
Plan: 13:30-13:50
Depend.
On time
Job 3000
AFT UD 5718VB 12M
Plan: 13:50-13:55
Pending
On time
Job 3300
AFT UD 5718VB 12M
Plan: 13:55-14:30
Depend.
On time
Job 3800
AFT UD 5718VB 12M
Plan: 14:30-14:40
Depend.
On time
Job 3000
AFT MC CADCON 12P
Pending
On time
13:19
view
13:00
13:30
13:50
13:55
14:00
Job: 3300
FWD LD 5460VB 12P
Plan: 13:00-13:30
Halted
On time
Job: 3800
FWD LD 5460VB 12P
Plan: 13:30-13:50
Depend.
On time
Job: 3000
Pending
Job: 3000
AFT UD 5718VB 12M
Plan: 13:55-14:30
Depend.
On time
13:19
JOB 3000
AFT UD 5718VB 12M (10000459589)
Start: 13:50 (Today) End: 13:55 (Today)
Pending
On time
now 13:50
13:55
Plan. duration.: 8 i.m.
Real duration: 0 i.m.
Station: V90
Work Ceter: C3 (Assembly)
Section: 16/19
Deck: Upper
MSN: MSN 12 UAE01-10
Material:
3x Screws M3 30mm
3x Nut M3
Tools:
Screw driver - Cross
Details
Start
Refuse

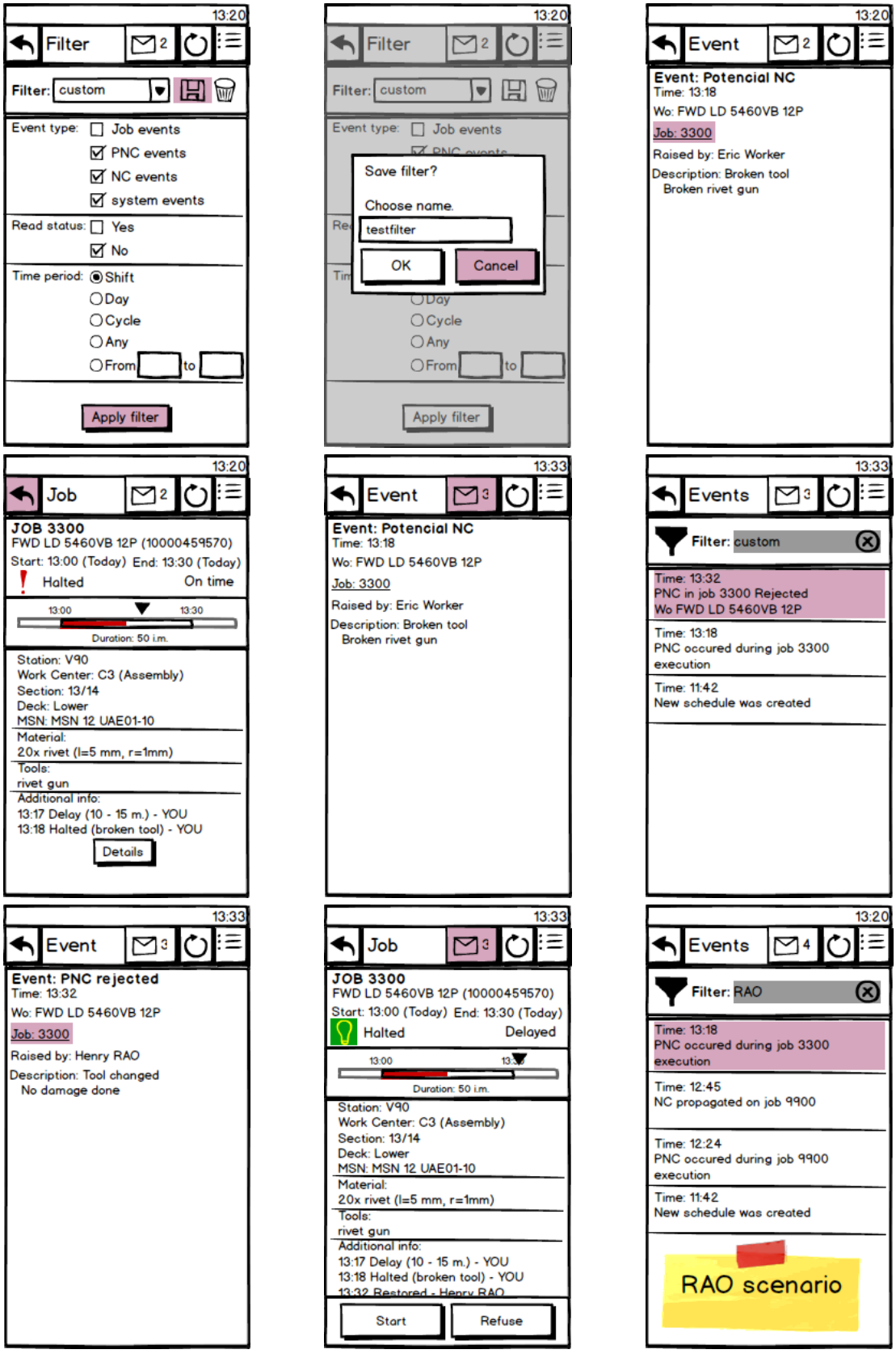

13:19
JOB 3000
AFT UD 5718VB 12M (10000459589)
Start: 13:50 (Today) End: 13:55 (Today)
Pending On time
Do You want to START this JOB?
Yes
No
Tools:
Screw driver - Cross
Details
Start
Refuse
13:26
JOB 3000
AFT UD 5718VB 12M (10000459589)
Start: 13:50 (Today) End: 13:55 (Today)
Running On time
Plan. duration.: 8 i.m. Real duration: 12 i.m.
Station: V90
Work Ceter: C3 (Assembly)
Section: 13/14
Deck: Upper
MSN: MSN 12 UAE01-10
Material:
3x Screws M3 30mm
3x Nut M3
Tools:
Screw driver - Cross
Details
Finish
Delay/PNC
Do You want to FINISH work on this JOB?
The job is completed.
Write the reason why You finish uncopleted job. (e.g. End of shift)
Yes
No
EVENTS
EVENT 1 (Time: 12:39)
JOB: 3000
WO: FWD MC XXXYYY ZZZ
Announcement: PNC resolved
Issued by: John Process
EVENT 2 (Time: 13:22)
JOB: 3300
WO: FWD LD 5460VB 12P
Announcement: PNC confirmed as NC
Issued by: John Process
JOB 3300
FWD LD 5460VB 12P (10000459570)
Start: 13:00 (Today) End: 13:30 (Today)
Terminated On time
Plan. duration.: 50 i.m. Real duration: 23 i.m.
Station: V90
Work Center: C3 (Assembly)
Section: 13/14
Deck: Lower
MSN: MSN 12 UAE01-10
Material:
20x rivet (l=5 mm, r=1mm)
Tools:
rivet gun
Additional info:
13:17 Delay (10 - 15 m.) - YOU
13:18 Halted (broken tool) - YOU
Continue
view
Job 3000 FWD LD 5460VB 12P Plan: 12:00-13:00 Finished Delayed
Job 3300 FWD LD 5460VB 12P Plan: 13:00-13:30 Terminat Delayed
Job 3800 FWD LD 5460VB 12P Plan: 13:30-13:50 Depend. Delayed
Job 3000 AFT UD 5718VB 12M Plan: 13:50-13:55 Running On time
Job 3300 AFT UD 5718VB 12M Plan: 13:55-14:30 Depend. On time
Job 3800 AFT UD 5718VB 12M Plan: 14:30-14:40 Depend. On time
Job 3000 AFT MC CADCON 12P Pending On time
Job: 3300 FWD LD 5460VB 12P Plan: 13:00-13:30 Terminat Delayed
Job: 3800 FWD LD 5460VB 12P Plan: 13:30-13:50 Depend. Delayed
Job: 3000 Running
Job: 3000 AFT UD 5718VB 12M Plan: 13:55-14:30 Depend. On time
13:32
JOB 3000
AFT UD 5718VB 12M (10000459589)
Start: 13:50 (Today) End: 13:55 (Today)
Finished On time
Plan. duration.: 8 i.m. Real duration: 20 i.m.
Station: V90
Work Ceter: C3 (Assembly)
Section: 13/14
Deck: Upper
MSN: MSN 12 UAE01-10
Material:
3x Screws M3 30mm
3x Nut M3
Tools:
Screw driver - Cross
Details
Finish
Delay/PNC
JOB 3300
FWD LD 5460VB 12P (10000459570)
Start: 13:00 (Today) End: 13:30 (Today)
Halted Delayed
Plan. duration.: 50 i.m. Real duration: 23 i.m.
Station: V90
Work Center: C3 (Assembly)
Section: 13/14
Deck: Lower
MSN: MSN 12 UAE01-10
Material:
20x rivet (l=5 mm, r=1mm)
Tools:
rivet gun
Additional info:
13:17 Delay (10 - 15 m.) - YOU
13:18 Halted (broken tool) - YOU
Continue

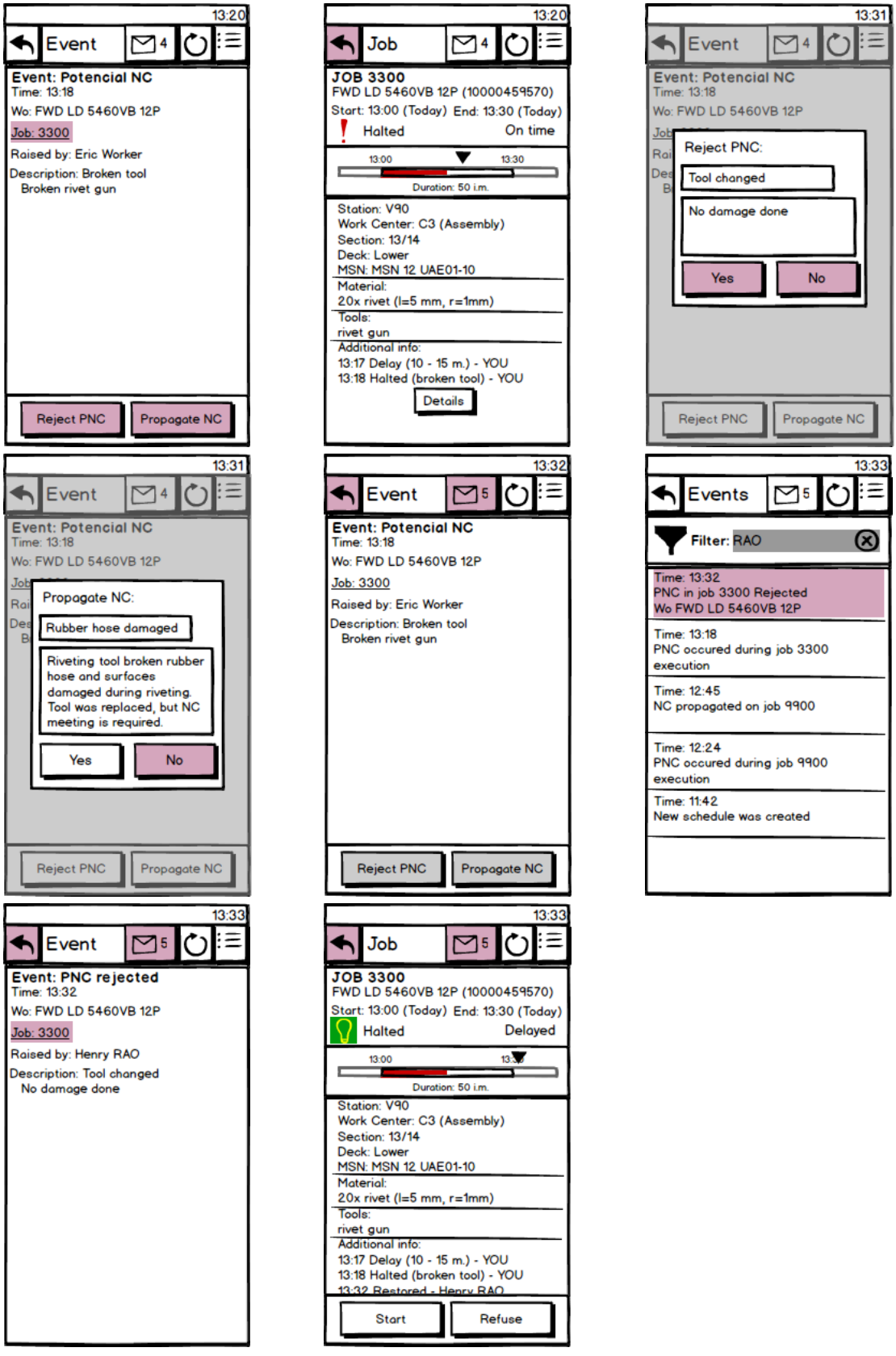

13:32
JOB 3300
FWD LD 5460VB 12P (10000459570)
Start: 13:00 (Today) End: 13:30 (Today)
Halted
Delayed
Do You want to CONTINUE this JOB?
Yes
No
rivet gun
Additional info:
13:17 Delay (10 - 15 m.) - YOU
13:18 Halted (broken tool) - YOU
Continue
13:32
view
Job 3000 FWD LD 5460VB 12P Plan: 12:00-13:00 Finished Delayed
Job 3300 FWD LD 5460VB 12P Plan: 13:00-13:30 Halted Delayed
Job 3800 FWD LD 5460VB 12P Plan: 13:30-13:50 Depend. Delayed
Job 3000 AFT UD 5718VB 12M Plan: 13:50-13:55 Finished On time
Job 3300 AFT UD 5718VB 12M Plan: 13:55-14:30 Pending On time
Job 3800 AFT UD 5718VB 12M Plan: 14:30-14:40 Depend. On time
Job 3000 AFT MC CADCON 12P Pending On time
13:32
Job: 3300 FWD LD 5460VB 12P Plan: 13:00-13:30 Halted Delayed
Job: 3800 FWD LD 5460VB 12P Plan: 13:30-13:50 Depend. Delayed
Job: 3000 Finished
Job: 3000 AFT UD 5718VB 12M Plan: 13:55-14:30 Pending On time
13:32
EVENTS
EVENT 1 (Time: 12:39)
JOB: 3000
W.O.: FWD MC XXXYYY ZZZ
Announcement: PNC disapproved
Issued by: John Process
EVENT 2 (Time: 13:22)
JOB: 3300
W.O.: FWD LD 5460VB 12P
Announcement: PNC confirmed as NC
Issued by: John Process
EVENT 3 (Time: 13:32)
JOB: 3300
W.O.: FWD LD 5460VB 12P
Announcement: NC resolved
Issued by: John Process
13:40
JOB 3300
FWD LD 5460VB 12P (10000459570)
Start: 13:00 (Today) End: 13:30 (Today)
Running
Delayed
Plan. duration.: 50 i.m. Real duration: 36 i.m.
Station: V90
Work Center: C3 (Assembly)
Section: 13/14
Deck: Lower
MSN: MSN 12 UAE01-10
Material:
20x rivet (l=5 mm, r=1mm)
Tools:
rivet gun
Additional info:
13:17 Delay (10 - 15 m.) - YOU
13:18 Halted (broken tool) - YOU
Finish
Delay/PNC
13:40
Do You want to FINISH work on this JOB?
The job is completed.
End of shift
Yes
No
13:41
JOB 3300
FWD LD 5460VB 12P (10000459570)
Start: 13:00 (Today) End: 13:30 (Today)
Pending
Delayed
Plan. duration.: 50 i.m. Real duration: 37 i.m.
Station: V90
Work Center: C3 (Assembly)
Section: 13/14
Deck: Lower
MSN: MSN 12 UAE01-10
Material:
20x rivet (l=5 mm, r=1mm)
Tools:
rivet gun
Additional info:
13:17 Delay (10 - 15 m.) - YOU
13:18 Halted (broken tool) - YOU
Continue
13:41
There is new schedule. Do you want to REFRESH?
Yes
No
13:41
EVENTS
EVENT 1 (Time: 12:39)
JOB: 3000
W.O.: FWD MC XXXYYY ZZZ
Announcement: PNC disapproved
Issued by: John Process
EVENT 2 (Time: 13:22)
JOB: 3300
W.O.: FWD LD 5460VB 12P
Announcement: PNC confirmed as NC
Issued by: John Process
EVENT 3 (Time: 13:32)
JOB: 3300
W.O.: FWD LD 5460VB 12P
Announcement: NC resolved
Issued by: John Process
EVENT 4 (Time: 13:41)
SCHEDULE:
Announcement: rescheduled
Issued by: Peter Stationmanager


13:41
4
view
Job 3000
FWD LD 5460VB 12P
Plan: 12:00-13:00
Finished
Delayed
Job 3300
FWD LD 5460VB 12P
Plan: 13:00-13:30
Pending
Delayed
Job 3800
FWD LD 5460VB 12P
Plan: 13:30-13:50
Depend.
Delayed
Job 3000
AFT UD 5718VB 12M
Plan: 13:50-13:55
Finished
On time
Job 3300
AFT UD 5718VB 12M
Plan: 13:55-14:30
Pending
On time
Job 3800
AFT UD 5718VB 12M
Plan: 14:30-14:40
Depend.
On time
Job 3000
AFT MC CADCON 12P
Pending
On time
There is new schedule.
Do you want to
REFRESH?
Yes
No
Refresh
Schedule
Actual job
Events
Dashboard
Logout
Do You want to Log out?
Job: 3300
Job: 3800
Job: 3000
13:00
13:30
13:50
13:55
14:00
14:30
14:40

Mock-ups Worker-Team Leader




13:20
Filter
2
Filter: custom
Event type: Job events
PNC events
NC events
system events
Read status: Yes
No
Time period: Shift
Day
Cycle
Any
From to
Apply filter
13:20
Filter
2
Filter: custom
Event type: Job events
Save filter?
Choose name.
testfilter
OK
Cancel
Day
Cycle
Any
From to
Apply filter
13:20
Event
2
Event: Potencial NC
Time: 13:18
Wo: FWD LD 5460VB 12P
Job: 3300
Raised by: Eric Worker
Description: Broken tool
Broken rivet gun
13:20
Job
2
JOB 3300
FWD LD 5460VB 12P (10000459570)
Start: 13:00 (Today) End: 13:30 (Today)
Halted
On time
13:00
13:30
Duration: 50 i.m.
Station: V90
Work Center: C3 (Assembly)
Section: 13/14
Deck: Lower
MSN: MSN 12 UAE01-10
Material:
20x rivet (l=5 mm, r=1mm)
Tools:
rivet gun
Additional info:
13:17 Delay (10 - 15 m.) - YOU
13:18 Halted (broken tool) - YOU
Details
13:33
Event
3
Event: Potencial NC
Time: 13:18
Wo: FWD LD 5460VB 12P
Job: 3300
Raised by: Eric Worker
Description: Broken tool
Broken rivet gun
13:33
Events
3
Filter: custom
Time: 13:32
PNC in job 3300 Rejected
Wo FWD LD 5460VB 12P
Time: 13:18
PNC occured during job 3300
execution
Time: 11:42
New schedule was created
13:33
Event
3
Event: PNC rejected
Time: 13:32
Wo: FWD LD 5460VB 12P
Job: 3300
Raised by: Henry RAO
Description: Tool changed
No damage done
13:33
Job
3
JOB 3300
FWD LD 5460VB 12P (10000459570)
Start: 13:00 (Today) End: 13:30 (Today)
Halted
Delayed
13:00
Duration: 50 i.m.
Station: V90
Work Center: C3 (Assembly)
Section: 13/14
Deck: Lower
MSN: MSN 12 UAE01-10
Material:
20x rivet (l=5 mm, r=1mm)
Tools:
rivet gun
Additional info:
13:17 Delay (10 - 15 m.) - YOU
13:18 Halted (broken tool) - YOU
13:32 Restored - Henry RAO
Start
Refuse
13:20
Events
4
Filter: RAO
Time: 13:18
PNC occured during job 3300
execution
Time: 12:45
NC propagated on job 9900
Time: 12:24
PNC occured during job 9900
execution
Time: 11:42
New schedule was created
RAO scenario


13:20
Event
4
Event: Potencial NC
Time: 13:18
Wo: FWD LD 5460VB 12P
Job: 3300
Raised by: Eric Worker
Description: Broken tool
Broken rivet gun
Reject PNC
Propagate NC
13:20
Job
4
JOB 3300
FWD LD 5460VB 12P (10000459570)
Start: 13:00 (Today) End: 13:30 (Today)
Halted
On time
13:00
13:30
Duration: 50 i.m.
Station: V90
Work Center: C3 (Assembly)
Section: 13/14
Deck: Lower
MSN: MSN 12 UAE01-10
Material:
20x rivet (l=5 mm, r=1mm)
Tools:
rivet gun
Additional info:
13:17 Delay (10 - 15 m.) - YOU
13:18 Halted (broken tool) - YOU
Details
13:31
Event
4
Event: Potencial NC
Time: 13:18
Wo: FWD LD 5460VB 12P
Reject PNC:
Tool changed
No damage done
Yes
No
Reject PNC
Propagate NC
13:31
Event
4
Event: Potencial NC
Time: 13:18
Wo: FWD LD 5460VB 12P
Propagate NC:
Rubber hose damaged
Riveting tool broken rubber hose and surfaces damaged during riveting. Tool was replaced, but NC meeting is required.
Yes
No
Reject PNC
Propagate NC
13:32
Event
5
Event: Potencial NC
Time: 13:18
Wo: FWD LD 5460VB 12P
Job: 3300
Raised by: Eric Worker
Description: Broken tool
Broken rivet gun
Reject PNC
Propagate NC
13:33
Events
5
Filter: RAO
Time: 13:32
PNC in job 3300 Rejected
Wo FWD LD 5460VB 12P
Time: 13:18
PNC occured during job 3300 execution
Time: 12:45
NC propagated on job 9900
Time: 12:24
PNC occured during job 9900 execution
Time: 11:42
New schedule was created
13:33
Event
5
Event: PNC rejected
Time: 13:32
Wo: FWD LD 5460VB 12P
Job: 3300
Raised by: Henry RAO
Description: Tool changed
No damage done
13:33
Job
5
JOB 3300
FWD LD 5460VB 12P (10000459570)
Start: 13:00 (Today) End: 13:30 (Today)
Halted
Delayed
13:00
Duration: 50 i.m.
Station: V90
Work Center: C3 (Assembly)
Section: 13/14
Deck: Lower
MSN: MSN 12 UAE01-10
Material:
20x rivet (l=5 mm, r=1mm)
Tools:
rivet gun
Additional info:
13:17 Delay (10 - 15 m.) - YOU
13:18 Halted (broken tool) - YOU
Start
Refuse

Installation guide

This installation guide describes how to install application into device, how to start it and how to set up the application.

It is necessary to mention, that application is still in stadium of development. The setting screen, which may be visible in this section shows setting that is required for easier development and debugging.

The situation will be different in case of real use. The advanced settings will be set up by responsible IT specialist and worker using application will not be allowed to change them.

Installation

There are several ways how to install application into selected device, but only the simplest one will be described.

1. Copy the APK file, which may be found on enclosed CD/DVD, into memory of your device.

   That can be achieved by sending the file by e-mail, bluetooth, copying to SD card or by copying after connection of device and computer with USB cable.

2. Modify your Android's settings to allow the installation of applications from other sources.

   Under `Settings` -> `Application Settings` enable `Unknown Sources`.

   Or under `Settings` -> `(More)` -> `Security` -> `Device administration` enable `Unknown Sources`.

3. Find APK file in your device with use of file manager and click on the file icon.

4. Accept permission required by application.

5. Open the application.

Start of application

The mobile application will be started with click on icon showed on figure 27

Figure 27: Launcher

The login screen, which was proposed in section 3.2, should be launched after start, but there is no one. Security service, which should handle authentication into application, is still in development by cooperating partner. There is no reason to have this screen at this moment.

Application setup

After first start of application the user will see only empty screen where schedule should be.

Select the Settings from list after clicking on hardware menu button (if the device have one) or action menu button in right top corner of application. The Settings screen, which should look like screen on figure 28, will open.

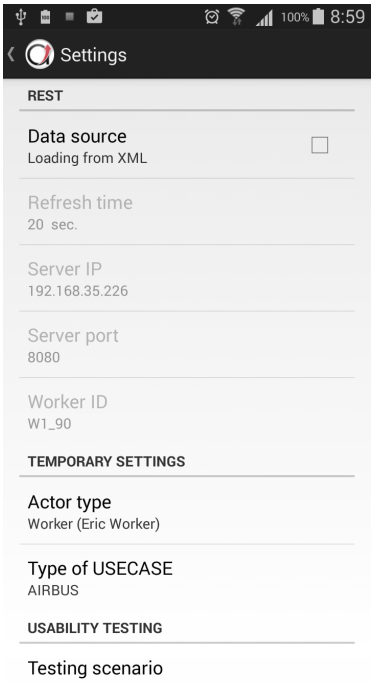


Figure 28: Application settings

The first selectable item define data should be provided by server with running client service (if there is one) or simulated data should be read from prepared XML file.

Loading data from web Client service/REST

Four items are enabled for further setting when the Client service/REST is selected as a data source. Beside items Server IP and Server port, which defines connection to

desired server, there are items Refresh time and Worker ID. Refresh time defines how often the application will request server for new data. The Worker ID item sets up identity of worker.

It has to be reminded that authentication of worker will be provided in different way by Security service, if the application is fully integrated into ARUM system.

Application requires allowed internet connection (Wireless or Mobile Data) in this case.

Loading data from XML

It is possible to select one of three prepared Testing scenarios when the XML file is selected as a data source. It means that it predefined jobs and events are provided. This setting simplify debugging and development of application and it provides easy way how to make usability testing repeatable.

Common temporary settings

The role of person (his competence), who works with application, is temporarily defined under item Actor type. Roles Worker, Team leader and Process manager may be selected. Worker and Team leader roles and their competences are described in sections 2.3,3.2. Role of Process manager will not be used in future implementation of application, however it is implemented now because of no existing tool covering his activities. Type of use-case defines whether the application is used in Airbus or in Iacobucci. This setting defines how states of jobs will evolve during use of application. Possible evolutions of states are presented in the figures 17.

This settings was required for usability testing. Both Actor type and Type of Use-case will be defined in different way, if the application is fully integrated into ARUM system.

Content of attached CD

```
CD
|
|--ArumWorker.apk
|--kovarpa8DIP.pdf
|
|--ArumWorker
|   |--libs
|   |--res
|   |--src
|   |--AndroidManifest.xml
|   |--ic_launcher-web.png
|   |--proguard-project.txt
|   |--project.properties
|
|--mockups
|
|--text
|   |--main.tex
|   |--appendices
|   |--bib
|   |--fig
|   |--src
|
|--testing_results
```