mConcAppt – A Mobile Interaction Design Method

Mobile business apps must be systematically engineered to guarantee high usability and user experience by construction. This guideline describes activities of a mobile UI and interaction design method (called mConcAppt) that has been developed by Fraunhofer IESE[[1]](#footnote-1). It is based on Fraunhofer IESE’s requirements engineering method called TORE (Task-oriented Requirements Engineering) which has been tailored to the domain of mobile business apps.

The mConcAppt method combines requirements engineering and UI & interaction design activities for mobile business apps and produces a so-called interaction concept as a basis for implementation and further activities.

The UI and interaction design must be closely interwoven with other engineering activities like architectural design to make sure the required usability and user experience can be achieved and to address trade-offs with other quality attributes besides usability and user experience that are of course important as well. This means for example that from a UI and interaction design standpoint the feature “live search” could be desired, but from the point of view of the architect the existing backend system does not provide the required performance and can only deliver the performance required for live search after cost intensive changes.

The app conception as a whole is an iterative process. Within the scope of the GSE Project, you will run through the following method phases:

|  |
| --- |
| * Phase 1: Elicit Requirements: comprises the gathering of requirements that are especially relevant for the interaction design (see section 1, page 2) * Phase 2: Specify Interaction Design: comprises the actual development of the interaction concept in tight collaboration with other software engineering disciplines (see section 2, page 13) * Phase 3: Validate Interaction Design: comprises validation of the interaction design with end users (see section 3, page 21) |

In the following sections, detailed guidance for each of the three phases is provided, including templates and examples of the artifacts to be produced during activities of each phase. The examples in this document are based on a project “method publication app” that aimed to implement an app that provides guidance to apply the mConcAppt method to interested stakeholders.

Furthermore, you can find further information:

* “TravelApp mConcAppt Interaction Concept Documentation.docx”: this document illustrates results of the application of mConcAppt on a concrete project example
* Android sketching templates
* Persona templates
* (Slides from lecture: Mobile Software Engineering - only given if you demand it)

# Phase 1: Elicit Requirements

The *Elicit Requirements* phase basically comprises all requirements engineering activities that have to be performed within the project. That is, the aim of the activities of this phase is primarily to prepare and conduct a requirements elicitation workshop and to document the elicited results.

For this purpose, the following tasks have to be conducted:

|  |
| --- |
| * Task 1.1: Prepare Requirements Elicitation Workshop (see section 1.1, page 2) * Task 1.2: Conduct Requirements Elicitation Workshop (see section 1.2, page 4) * Task 1.3: Document Elicited Requirements (Workshop Results) (see section 1.3, page 11) |

## Prepare Requirements Elicitation Workshop (Task 1.1)

The workshop preparation subsumes all activities that are necessary prior to conducting the workshop. This includes

|  |
| --- |
| * Definition of a Workshop Agenda * Selection of Participants * Initial Elicitation of Available Information |

In addition, the workshop responsible has to make sure, that needed material for workshop conduction is prepared (flip charts, beamer, pencil, cards), and that a whiteboard or something similar is available in the workshop room. Taking care of beverages and cookies is also obligatory to create a harmonious creative atmosphere in the workshop.

The **Workshop Agenda** should be aligned with the intended outcomes of the workshop (as summarized in Table I). That is, these outcomes include knowledge about:

Table I: Expected workshop outcomes

|  |
| --- |
| Expected Outcomes |
| relevant stakeholders, their roles in the project and their goals |
| characteristics of end users |
| as-is situations and current problems |
| to-be situations including activities that require mobile support |
| usage context |
| main system functions |

When preparing the agenda, you should allocate enough time to elicit the aforementioned information. For each outcome, there should be an item on the agenda. Table II below shows an example agenda for the workshop conduction that might be suitable for the workshop. But you might also adapt it.

Table II: Example workshop agenda

|  |  |
| --- | --- |
| Agenda | |
| Elicit information about involved stakeholders (including their roles and goals) and end user characteristics | 30 mins |
| Elicit as-is situations and current problems | 20 mins |
| Break | 10 mins |
| Elicit to-be situations including mobile activities and information about usage context | 30 mins |
| Elicit main system functions | 30 mins |
| Wrap-up and next steps | 10 mins |

Table III provides an overview on **participants that should typically be selected** as representatives of stakeholders and be invited to the workshop. In particular, the selection of appropriate (lead) users is essential for a successful workshop and reliable outcome. Please note, in this GSE course, the customer (Steffen Hess) will also represent the (lead) users.

Table III: Workshop Participants

|  |  |  |
| --- | --- | --- |
| Participant | Quantity | Role in the workshop |
| Interaction Designer & Requirements Expert | 1-2 | Workshop conception and preparation; Workshop moderator. |
| (Lead) User | 2-4 | Gives input from a user’s perspective. A user is a person who actually uses the system. The lead user approach should be chosen if no real users are available or if the system offers a high innovation potential.  “Lead users face needs that will be general in a marketplace – but face them months or years before the bulk of that marketplace encounters them, and lead users are positioned to benefit significantly by obtaining a solution to those needs.[[2]](#footnote-2)” |
| Project Management | 1 | Gives input from project management perspective. This role is fulfilled by the internal decision maker and responsible person for project conduction. |
| Business Analyst | 1 | Writes down minutes of the workshop (the interaction designer is usually not able to make complete notes about the given input because of moderating tasks); input from business perspective with regard to feasibility can be integrated directly into the minutes to not interrupt idea creativity during the workshop.  If the role business analyst and interaction designer is fulfilled by the same person – audio or video recording is mandatory to be able to recapture elicited information afterwards. |
| Customer | 1 | Gives input and constraints from a customer perspective. This role is fulfilled by the decision maker of the customer. This is usually the project management responsible person on customer’s side. |

In case that any relevant **information can be elicited offline** prior to the workshop (e.g., via document analysis), the information should be prepared and discussed during the workshop with all participating stakeholders. This discussion is important in order to add missing information, to overcome ambiguous information as well as to correct errors. For example, involved stakeholders, different user roles and user personas might be elicited before conducting the workshop.

In case of the GSE project, you might for example access related webpages of “people’s busses” (Bürgerbus), such as <http://www.weilerbach-pfalz.de/kulturelle-einrichtungen/buergerbus-weilerbach/index.html>

## Conduct Requirements Elicitation Workshop (Task 1.2)

The Requirements Elicitation Workshop should be conducted following the previously defined workshop agenda (see Table II). During the workshop, the moderator has to keep the discussion on track without interrupting the free flow of ideas and comments. The moderator also needs to ensure, that all workshop participants contribute to the discussion and encourage silent participants to contribute in a way that it is not possible that a single participant dominates the group. The moderator should try to overcome hierarchical structures – in general all roles are equal during workshop conduction.

The following sections provide further information about the elicitation of expected outcomes during the workshop (as summarized in Table I on page 2).

**Elicitation of stakeholders, their roles and goals:** the first step in the workshop is to i*dentify involved stakeholders* and to specify *their role* with the system. A stakeholder is every person influenced somehow by the system. Within the workshop, all identified stakeholders are assigned to a specific role (e.g. user, management of the system). Stakeholder name and role are not necessary distinguished. In addition, the workshop participants should elicit every stakeholder’s *main goals*.

Stakeholders can be elicited using the following template:

Table IV: Stakeholder Elicitation Template

|  |  |  |
| --- | --- | --- |
| Stakeholder Name | Stakeholder Role | Stakeholder Main Goals |
| Name | Role regarding the upcoming software system | 2-3 main goals of the stakeholder group |

Figure 1 shows examples of stakeholder descriptions that have an interest in the “method publication app”.

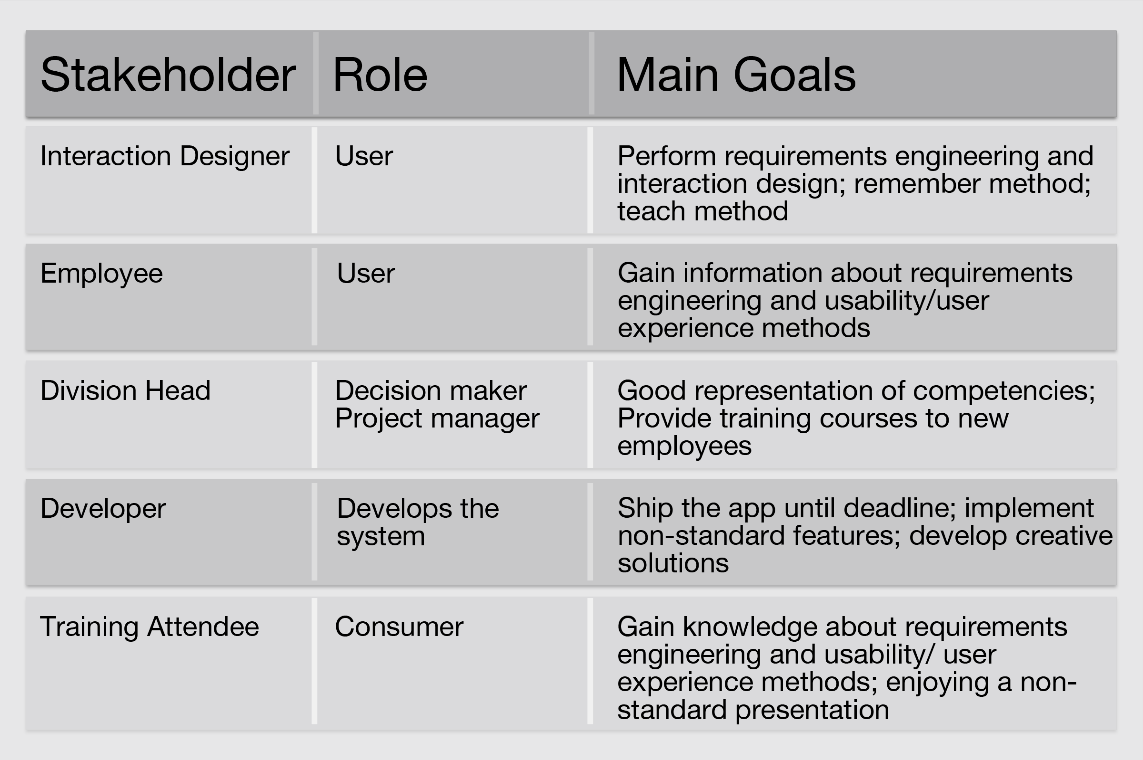


Figure 1: Elicitation of stakeholders example

**Elicitation of end user characteristics:** after the workshop, you should create at least 2 *user personas* (see also section 1.3, page 11). These user personas describe various goals and behavior patterns of potential end users of the system to be build.

The *User Persona* is an instantiation of the group of end user for one concrete example that comes close to the 80% of the user. There is no need for covering all possible exceptions – however, it is strongly recommended to focus on the main user and the main usage context of the system.

The user personas will represent the end user(s) in all upcoming discussions during in the project. That is, every upcoming decisions should be made keeping in mind if this decision fits to the created personas. Moreover, you might also use the persona descriptions to select possible user representatives for the purpose of validation activities (see section 3, page 21).

In order to create these personas, you should elicit and discuss detailed information about characteristics of end users during the workshop with representatives of the (lead) users. You might use the items listed in Table V as elicitation guideline (feel free to adapt this list of items to your project). Grey items do not necessarily need to be discussed in the workshop.

Table V: Elicitation of user characteristics to create personas

|  |  |
| --- | --- |
| Item | Description |
| Name | Concrete name of the persona |
| Picture | Concrete picture of the persona (can be taken from internal photo database) |
| Role | Role of the persona |
| Tagline | One sentence conclusion that represents the personas personality |
| Demographic data | Age, job position |
| Core characteristics | Core characteristics that helps to construct the personas personality |
| Core goals | Core goals of the persona to be accomplished |
| Typical challenges | Typical challenges of the persona while trying to achieve the core goals during work |
| Singularities | Special characteristics of the persona that differentiates him/her from other employees |
| Working situation | General working context of the persona (stress level, shift duration, tasks, etc.) |
| Working place | Distribution of work locations (e.g. 30% office, 70% on the road) |
| Expertise | General expertise of the persona according to software, IT |
| Main tasks with the system support | Main tasks, the persona will have to perform with the system |
| Most important tasks | Most important tasks the persona has to perform with or without system support |
| Least important tasks | Least important tasks the persona has to perform with or without system support |
| Miscellaneous | Miscellaneous information that helps make the persona alive (e.g. persona is an enthusiastic hobby pilot) |
| Prospective concrete usage context | Information about the context in which your system will typically be used. This might include information about locations of the user, physical environment (e.g. day times, lightning conditions), computing environment (e.g. devices, network capacities) etc. |

Figure 2 shows a concrete example of a persona that has been created during a workshop. It represents an interaction designer at Fraunhofer IESE who is one of the lead users for a method publication app.



Figure 2: User persona workshop example

**Elicitation of as-is situation and related problems:** Goal of the as-is elicitation is to derive detailed technical constraints and descriptions of the as-is situation within the project scope. The objective is to get a deep understanding of current processes, tasks, involved roles, data etc. in order to design suitable to-be situations in the subsequent discussion.

The as-is situation should be elicited together with problems that occur today. This step is done to get a common understanding of the actual situation before actually talking about any solution. Having the as-is situation described on a flipchart gives every participant of the workshop a starting basis for upcoming activities. Nevertheless, the major benefit of this description is to derive technical constraints and requirements within the project.

During description of the as-is situation the team should always focus on actual problems, that should be addressed or solved by the future system. These problems should be written down on an additional sheet. Solution concepts are not part of this description and the discussions. The as-is situation should be described step by step from the persona’s perspective following the template (see **Error! Reference source not found.**). The description should be in the form of concrete scenarios.

It is important to focus on step by step elicitation of the as-is process. Before going over to the next step, commitment of user to the previous step is required. The moderator should force participants to really name their major problems. This is often a problem especially when executives or high-ranked people are present in the workshop.

Table VI: Template to describe as-is situations

|  |  |
| --- | --- |
| Item | Description |
| Context | Context that leads to the fulfillment of the actual as-is scenario |
| Precondition | Precondition for scenario conduction |
| Step 1-N | Steps that are performed including occurring problems per step |
| Postcondition | State that is achieved after the scenario |

Figure 3 shows an excerpt of an example as-is situation description that has been discussed during the workshop for the development of a method publication app. It describes the situation in which the user persona talks to a prospective customer that is interested in mobile applications and especially the method for the conceptual design of them.

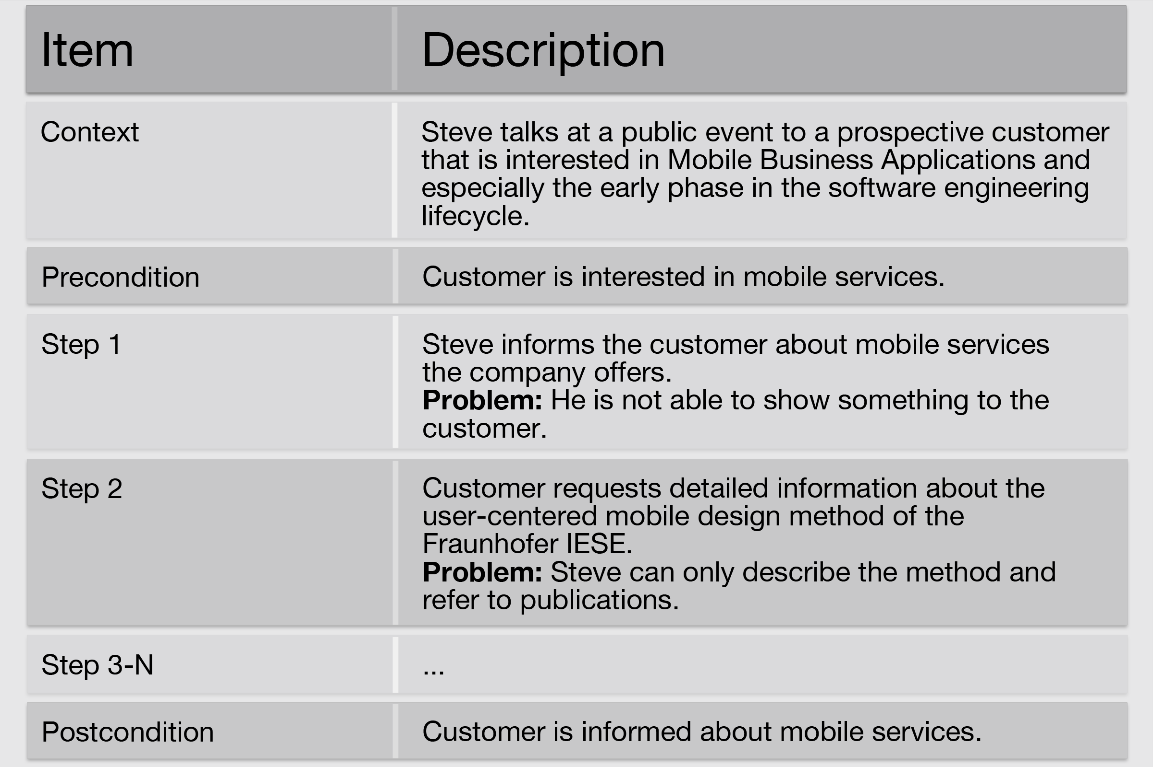


Figure 3: As-is situation description (excerpt) on the example of the method publication app

**Elicitation of to-be situation:** The to-be situation describes the usage of the system to be built in the future and should be elicited based on the as-is description derived in the previous step. That is, for each as-is situation, a corresponding to-be situation has to be elicited. The description itself should always be as complete as possible and key solution concepts can already be embedded in the description. Anyway, descriptions of the to-be situation should not focus on technical solution concept but rather describe the future system from a user perspective.

Stakeholder problems mentioned during the as-is elicitation should be assigned to goals solved by the to-be situation.

Table VII shows a template for description of to-be scenario that you can use as elicitation guideline during the workshop.

Table VII: Template to describe to-be situations

|  |  |
| --- | --- |
| Item | Description |
| Context | Describes the context that leads to the actual scenario – this can be the general usage context. Usually the overall situation has one general usage context and several tasks that are performed within the same context. |
| Precondition | Precondition for scenario conduction |
| Step 1-N | Steps that are performed – mark steps that are performed using the mobile device |
| Postcondition | State that is achieved after the scenario |

Analogous to the as-is elicitation, the moderator should focus on a step by step elicitation process. The workshop participants should determine system responsibilities and identify steps that need mobile assistance and also steps that not necessary need to be performed with mobile assistance. Steps of the to-be situation should be identified as:

* Human Activities: steps that are only performed by humans
* Human-System Activities: steps that are only performed by humans with system support
* System Activities: steps that are automatically performed by the system
* System-System Activities: steps that are performed by the system with support from external systems

During the workshop it is only necessary to mark activities identified for mobile support but the interaction designer should understand the way activities are identified for documentation. Usually, there is no need for performing every step on the mobile device. At this point in the workshop, there are no constraints – statements as “this is not possible with our backend system” should not be mentioned at all while eliciting the to-be situation. Nevertheless, the business analyst might integrate comments into the minutes about feasibility of described solutions. In addition, constructive solution proposals might be integrated as well. In practice, users often mention their needs without having to think about realization feasibility – therefore especially the business analyst has to understand those needs by making comments into the minutes about practical solution possibilities for these needs.

Figure 4 shows an example to-be situation description that has been discussed during the workshop for the development of the method publication app.

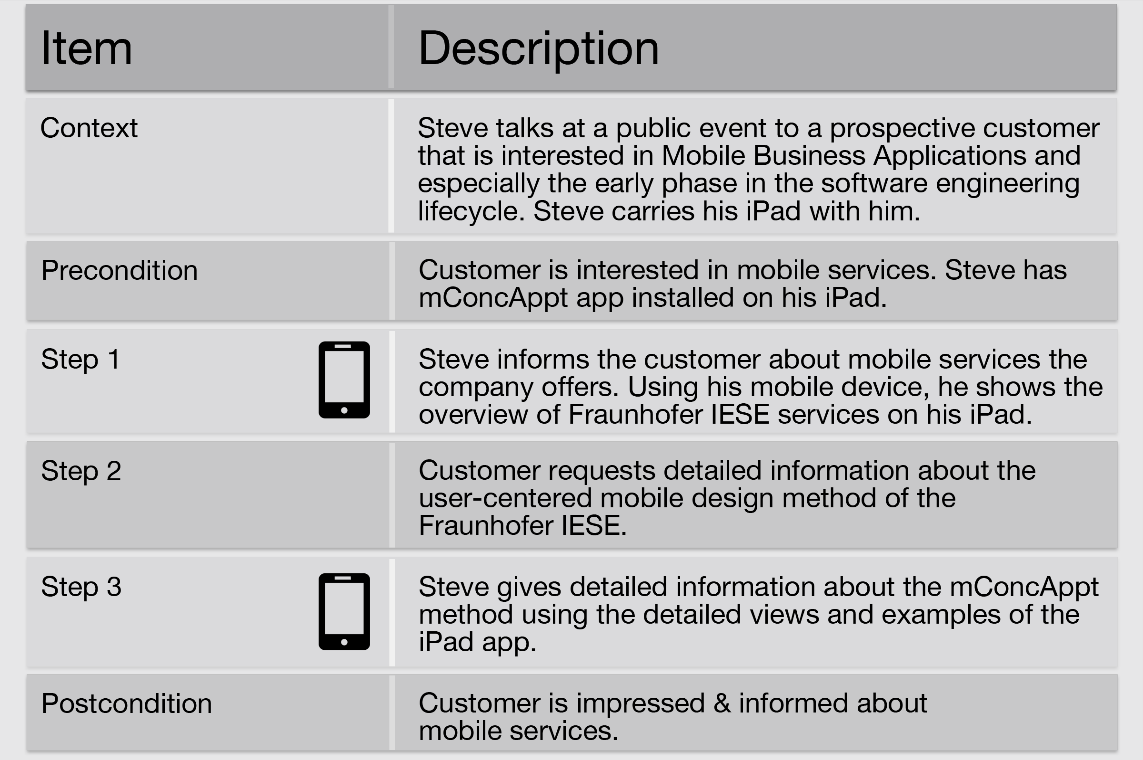


Figure 4: To-be situation workshop example

**Collection and discussion of relevant system functions: T**he identification of the main system functions completes the workshop. These system functions are based on the previously elicited to-be situations. In addition, exchanged data is identified with support of the business analyst who will be able to provide exact information about exchanged data and data formats based on this initial elicitation. The main system functions represent the core functionality of the system that has to be designed in the first iteration.

Within the workshop, it is sufficient to simply name the basic system functions to be able to define the scope of the app. It is sufficient to fill out ID, name and a short version of the description from the system function template (see Table VIII). Further details can be elaborated later on in the subsequent phase (see section 1.3, page 11).

Table VIII: System function description template

|  |  |
| --- | --- |
| Item | Description |
| ID | Unique identifier of the system function |
| Name | Name of the system function |
| Input Data | Data, that is entered into the system |
| Precondition | System precondition |
| Description | Step by step description of systems’ activities |
| Exception | Exceptions that might occur during the system functions |
| Business rules | Business rules that set constraints for the system function |
| Quality requirements | Quality requirements for the system function |
| Output data | Data, the system sends to the user(s) or other systems |
| Postcondition(s) | System postcondition(s) |

Figure 5 shows an example of a system function description that has been derived within the “method publication app”. The system function describes how detailed information to a specific part (step) of the method is displayed.

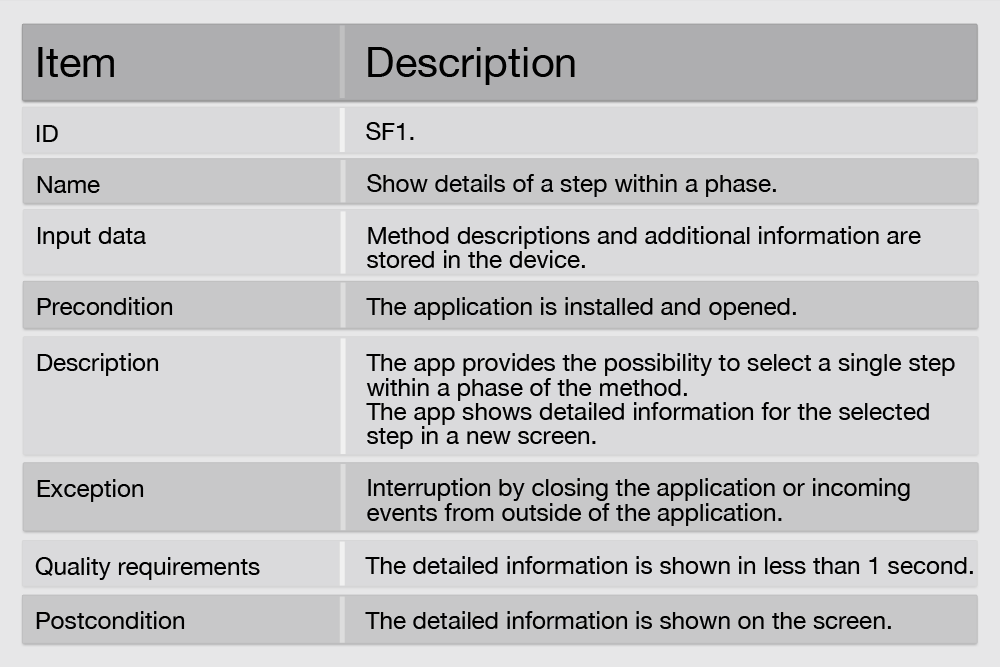


Figure 5: System function description example

**Wrap-up and Next Steps:** As a conclusion of the workshop, the next steps are written down and contact persons are named in case of upcoming questions. The moderator has to make sure to have contact persons to further clarify open issues regarding the following:

* Affected Stakeholders
* Stakeholder Goals
* Stakeholder Description
* User Persona
* Description of as-is Situation
* Problems in as-is Situation
* Description of to-be Situation
* Technical Constraints
* Exchanged Data

## Document Elicited Requirements (Workshop Results) (Task 1.3)

This task comprises processing and documentation of information gathered during the workshop in a form, that all involved project stakeholders are able to comprehend all elicited requirements information.

The resulting workshop documentation represents the requirements specification (and is part of the interaction concept description) and forms a basis for all upcoming activities. It can be seen as a first draft version of the concept and documents all elicited information in a structure (see Table IX) that is similar to the workshop agenda.

It is a lightweight documentation, focusing on the information that is needed for further steps.

**Workshop Documentation Template:** Table IX shows a structure template that you might use to create the requirements document. The template includes a description of every chapter and if applicable a related link to the information elicited during the workshop and corresponding description templates.

Table IX: Workshop documentation template

|  |  |  |
| --- | --- | --- |
| No | Chapter name | Description of chapter content |
| 1 | Introduction | Short description of the document; Contents; Reason and purpose of the document; Text can be very generic and reusable but delivers a good starting point if someone reads the document for the first time. |
| 2 | Usage context | The usage context is a plausible description of the environment and situation based on a coherent set of assumptions. Describes the 95% case when the system will be used. The usage context does not include specific solutions.  Usage context is elicited during the workshop when discussing end user characteristics / personas (see Table V), the as-is situation (see Table VI) and to-be situation (Table VII). |
| 3 | Stakeholders and goals | Short description of involved stakeholders and their role and goals.  Derives directly from the given template (Table IV) – might be extended by additional information from clarifying talks. |
| 4 | User persona | Description of the user persona, if more than one typical user role is available – each user role should be represented by one user persona.  Derives directly from the information elicited in the given template (Table V). It is recommended to use a more illustrative representation as shown in the example chapter (see Figure 2). You can find templates that you can use and/or adapt. |
| 5 | As-is situation | Detailed and complete description of the actual situation. Often in scenarios might also include actual problems. No solution concepts. Description from user perspective mandatory.  Derives from elicited information using the given template (Table VI). It is useful to instantiate the more broad general usage context elicited in the workshop for every concrete situation in detail to capture the specific detailed usage context for every situation. |
| 6 | Problems in as-is situation | Problems discovered during analysis of as-is situation are listed explicitly in this chapter.  It is sufficient to simply list them. However, they might be structured according to specific stakeholder problems if applicable. |
| 8 | To-be situation | Complete descriptions of the to-be situation including usage of the upcoming system. Description of all different usage scenarios. Already embedding of possible solution concepts. Description from user perspective mandatory.  Derives from elicited information using the given template (Table VII). Again instantiate the more broad general usage context elicited in the workshop for every concrete situation in detail. If it turns out that there are more than 4 completely different usage contexts within on single app the interaction designer should discuss with the project management if this should really be one single app – or if a portfolio with different apps for completely different usage context might be more suitable. |
| 9 | Main system functions | Complete description of system functions using the proposed template (see Table VIII). One key solution concept can be realized by more than one system function. This step is important for preparing the app development and the actual interaction design. At this point in time main system functions should be seen as a high level description of the actual systems’ functionality. |
| 10 | Constraints | Constraints sections should be subdivided into technical constraints and project constraints.  Technical constraints (e.g. selected device, device properties, given technical environment (backend systems) or procedures, UI guidelines) usually derive from overview talks with business experts, developers, architects and as-is descriptions. In case of disagreement - project management has to decide which way to go. In case of constraints there is often no right or wrong and both positions usually have good reasons for their opinions. Nevertheless constraints are limiting the opportunities and flexibility of the interaction concept.  Project constraints (e.g. time restrictions; budget; legal constraints, number of planned increments) set the boundary for project conduction and often limit available resources and realization opportunities.  Both technical and project constraints might be extended afterwards based on additional input from other project stakeholders. |
| 11 | Next steps | Documentation of upcoming activities and responsible persons derived from the wrap-up session at the end of the workshop. |

# Phase 2: Specify Interaction Design

The specification of the interaction design (see **Error! Reference source not found.**) comprises the construction of the actual interaction concept, without consideration of the underlying backend system in the first iteration. The Mobile Business App is specified based on information elicited during the initial *Elicit Requirements* phase (see section 1, page 2). Within the phase, the following tasks should be executed:

|  |
| --- |
| * Task 2.1: Identify key functionality (~1h) (see section 2.1, page 13) * Task 2.2: Specify interaction cases (~30 mins per interaction case) (see section 2.2, page 13) * Task 2.3: Model flow of interaction cases (~15 mins) (see section 1.3, page 11) * Task 2.4: Create wireframes (~5-30 mins per wireframe iteration) (see section 2.4, page 16) * Task 2.5: Model screen flow (20 mins) (see section 2.5, page 19) |

First, the *Key Functionalities* are derived from the to-be situations. These *Key Functionalities* form the starting point of the app and should be represented by the first specified *Interaction Cases*. During the specification of the *Interaction Cases*, the *Flow of Interaction Cases* is assembled step by step. After this specification, *Wireframes* are created based on the *Interaction Cases*. Analogous to the *Flow of Interaction Cases*, the actual *Screen Flow* is assembled. The time estimation for each tasks (see list above) are rough estimations based on observations during applications of mConcAppt in real projects. Please note: Especially specification of *Interaction Cases* and creation of *Wireframes* are creative steps and might take more time than estimated. Providing an interaction concept of high quality is more important than sticking to time parameters.



## Identify Key Functionality (Task 2.1)

Take the to-be scenario of the workshop and identify key functionalities. Key functionalities are tasks that really have to be performed mobile in the given context. There is always a huge risk to come up with too many functionalities based on existing desktop or legacy systems. Ask yourself: “What needs really to be performed on the mobile device”.

Key functionalities are marked in the to-be situation description (see Table VII).

## Specify Interaction Cases (Task 2.2)

Take the identified *to-be scenarios* including identified *key solution concepts* and *key functionalities* to derive Interaction Cases (use cases) from that information. According to agile development processes, one *Interaction Case* comprises several user stories that are related based on a certain task. Usually exactly one human action and one system action (see Table X) should form a user story.

First, identify human-system interactions, human-human interactions and system-system interactions in the to-be scenario while system means in this case the upcoming mobile app. Then take all human-system interactions and write them down in form of interactions - look after similarities or possible cases that can be combined to one *Interaction Case*.

Than describe each *Interaction Case* using the template given below (see Table X), which is very similar to use cases but focusses on information needed to specify the app prototype.

When creating the *Interaction Cases* there are no concrete design solutions done rather than showing the concept of user interaction. Concrete interaction design might be done by someone else based on your information. In case the system is used by different user groups – the user group performing the *Interaction Case* has to be specified as well.

Table X: Interaction case description template

|  |  |
| --- | --- |
| Item | Description |
| ID | Unique identifier of the interaction case |
| Usage Context | The usage context is a plausible description of the environment and situation based on a coherent set of assumptions. Interaction case usage context is a particular instantiation of the more general usage context that is described in the to-be situation description (see Table VII) |
| Human Action 1 | A human action is an action a human performs on the corresponding systems’ input device. If applicable, each human action is further described by the concrete usage type (e.g. tap, double tap, speech input, pinch). In addition exchanged data between user and system is named. |
| System Action 1 | A system action is the action the system performs in order to present the intended output to the user. System action description is extended by describing input feedback as well as media and modalities for the system action (e.g. confirm input via change of button color and auditory notification and present system response using an animation with text. Screen transitions are also described here. |
| … | … |
| Human Action N | (see description of human action 1) |
| System Action N | (see description of system action 1) |
| Postcondition(s) | The post condition explicitly shows where the user goes in the system after performing this interaction case. |

Figure 6 shows a concrete example of an interaction case description from the method publication app.

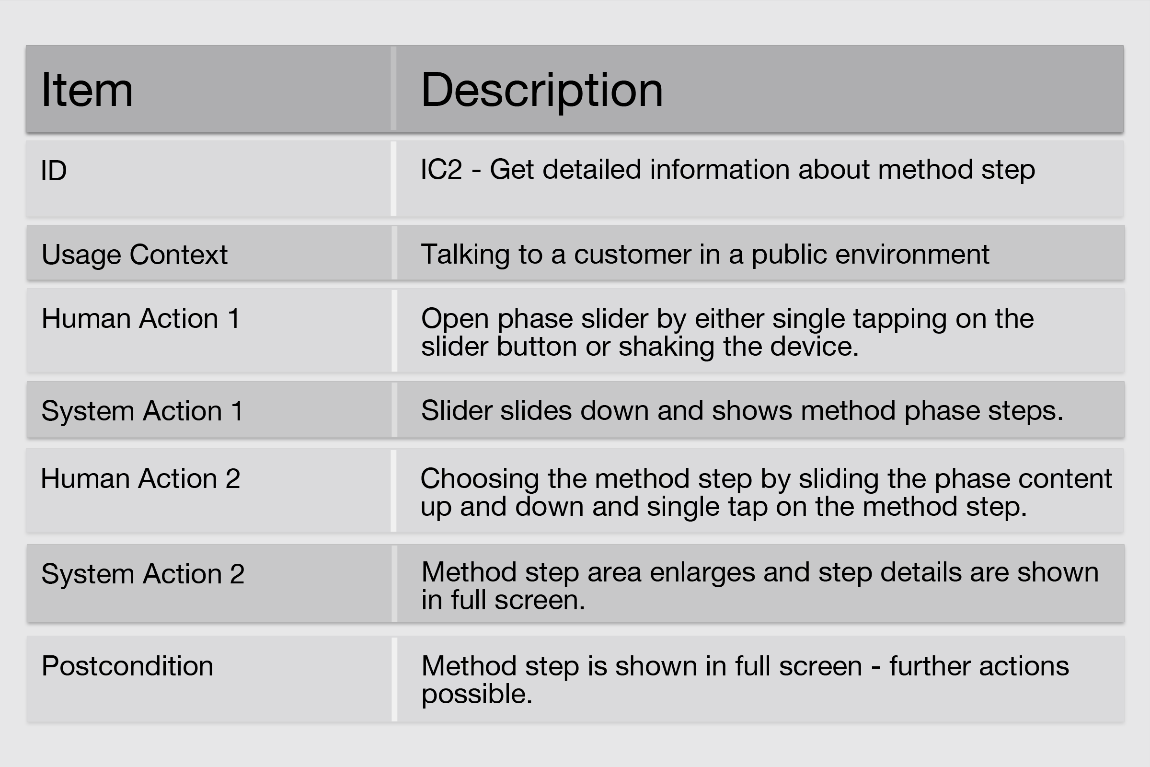


Figure 6: Interaction case example

## Model Flow of Interaction Cases and Their Relationship (Task 2.3)

In this step, all *Interaction Cases* are shown in a graphic representation (see Figure 7) using the unique identifiers given in the descriptions of the interaction cases (see section 2.2). This graphic representation allows getting a quick overview about the general structure and function range of the app.

Based on the *Flow of Interaction Cases* the final decision about the scope of the app is done by the interaction responsible in agreement with the project management.

Figure 7 shows an example of an *Interaction Case Flow*. Arrows show the straight forward user flow between different interaction cases. Swim lanes might also indicate logically divided app contents.

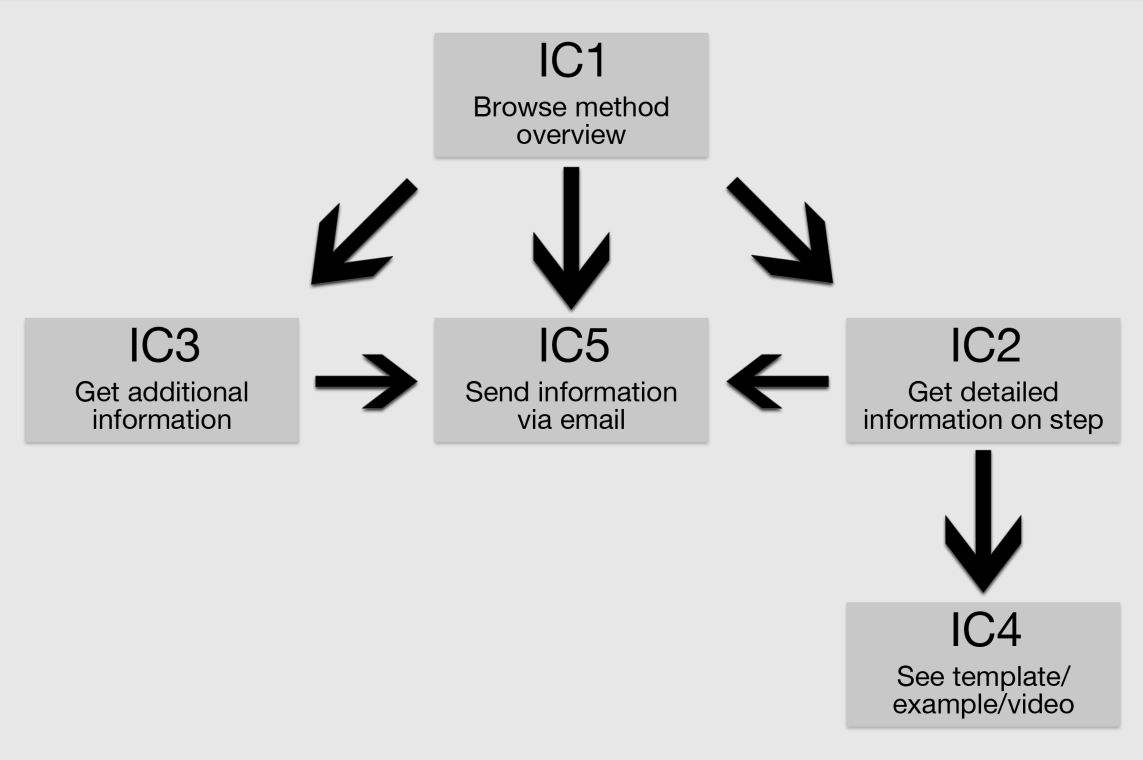


Figure 7: Flow of interaction cases example

## Create Wireframes (Task 2.4)

Based on the *Interaction Cases* (see section 2.2) and the *Flow of Interaction Cases* (see section 2.3) the actual *Wireframes* are created.

In first iterations, *Wireframes* are created using paper and pencil. This is the quickest way to create several versions of a screen and to try out different solutions. To guarantee traceability, wireframes should be created on a template (similar to Figure 8) that shows *name of the project*, interaction case & screen identifier (unique ID based on interaction case), version of the wireframe, and date of production.

Please note: you can find sketching templates for the android app that you can use to create your own templates to create pen & paper wireframes.

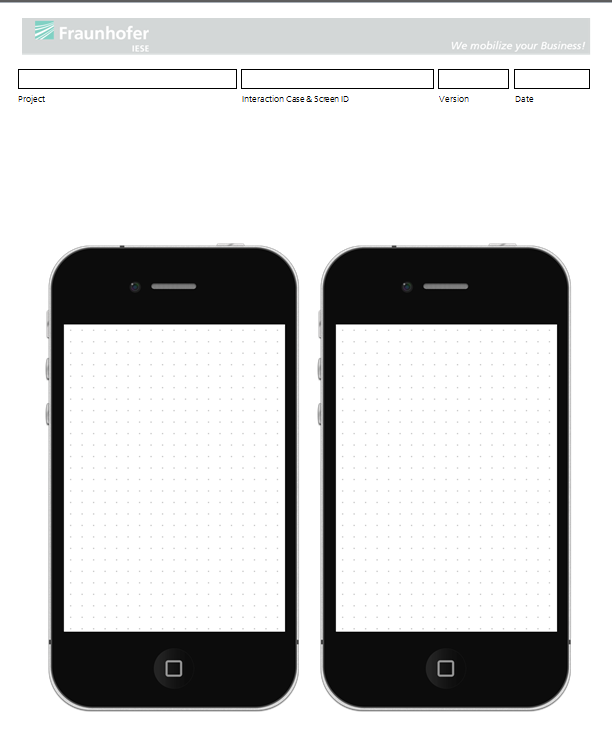


Figure 8: iPhone wireframe template

Based on the paper & pencil wireframes, Powerpoint / Keynote prototypes are produced prior to the first review process. Templates for Powerpoint and Keynote prototyping are for example available at Keynotopia[[3]](#footnote-3). Depending on the designers preference even more high fidelity tools (e.g. Adobe Photoshop, Balsamiq Mockups) can be used.

Figure 9 to Figure 12 show several versions of the same screen to demonstrate how the prototype evolves during several iterations of wireframing. Figure 13 shows the according high fidelity version that is very close to the final design and already integrates the visual design aspects.



Figure 9: Very early concept sketch

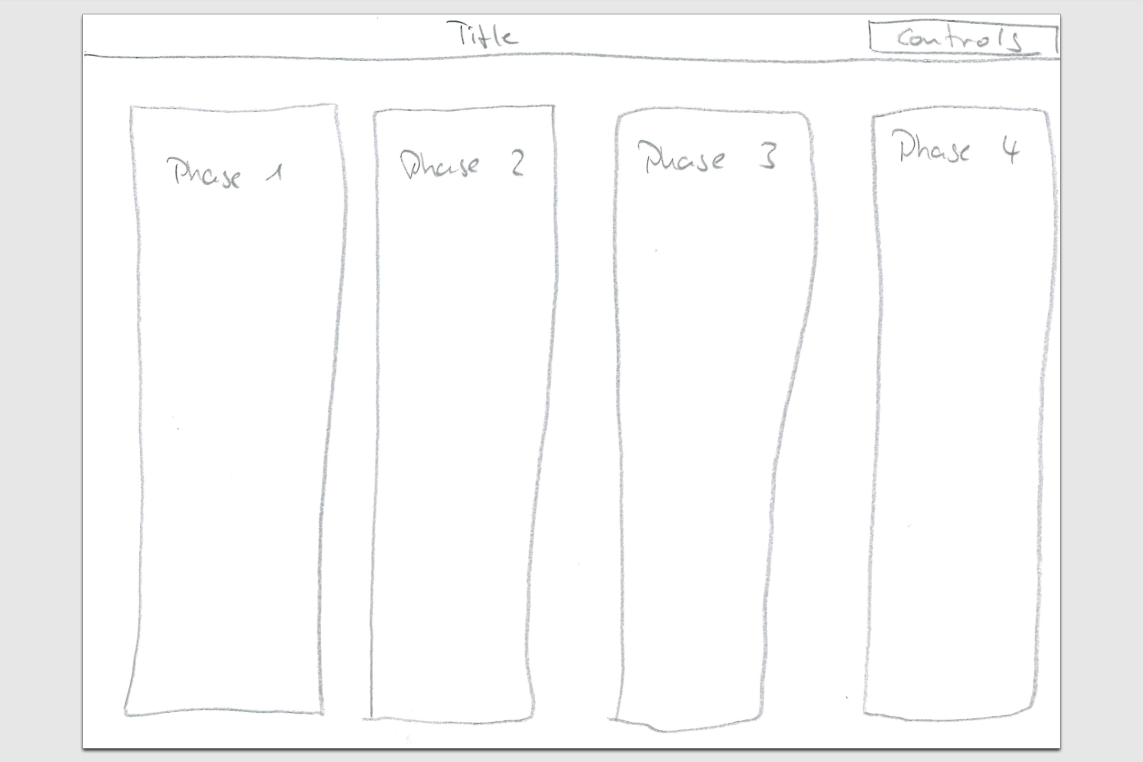


Figure 10: Second version of early concept sketch

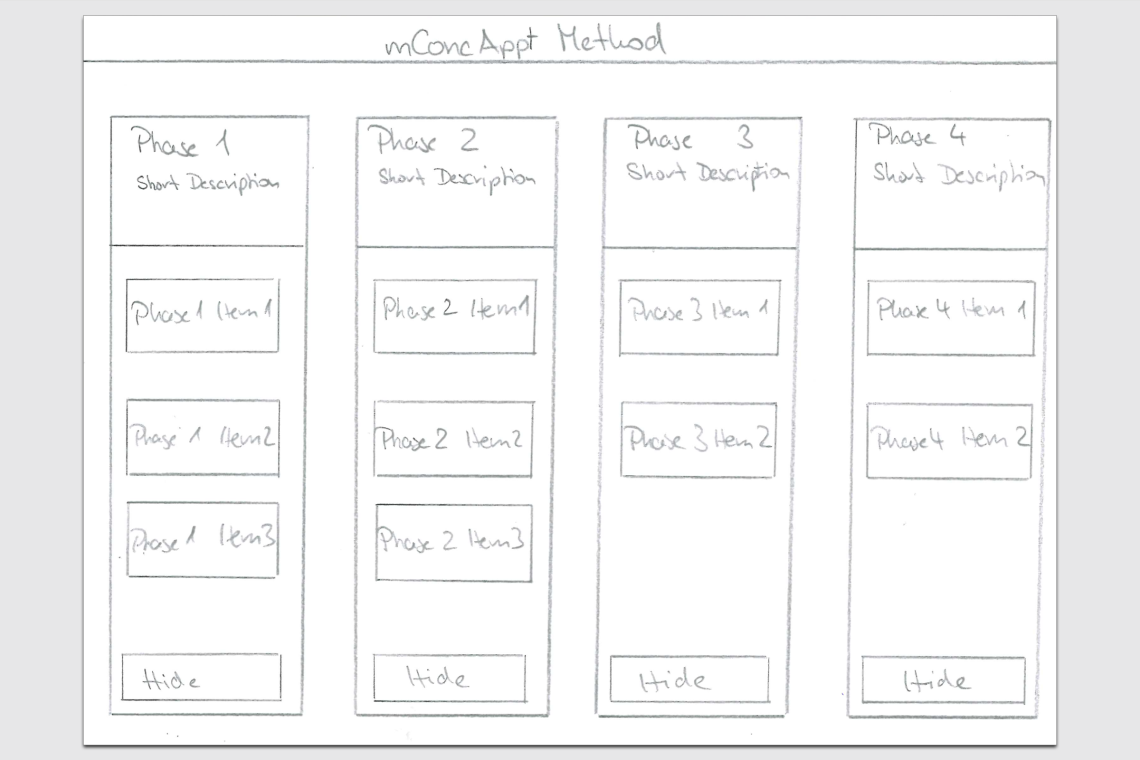


Figure 11: Third version shows already some screen details

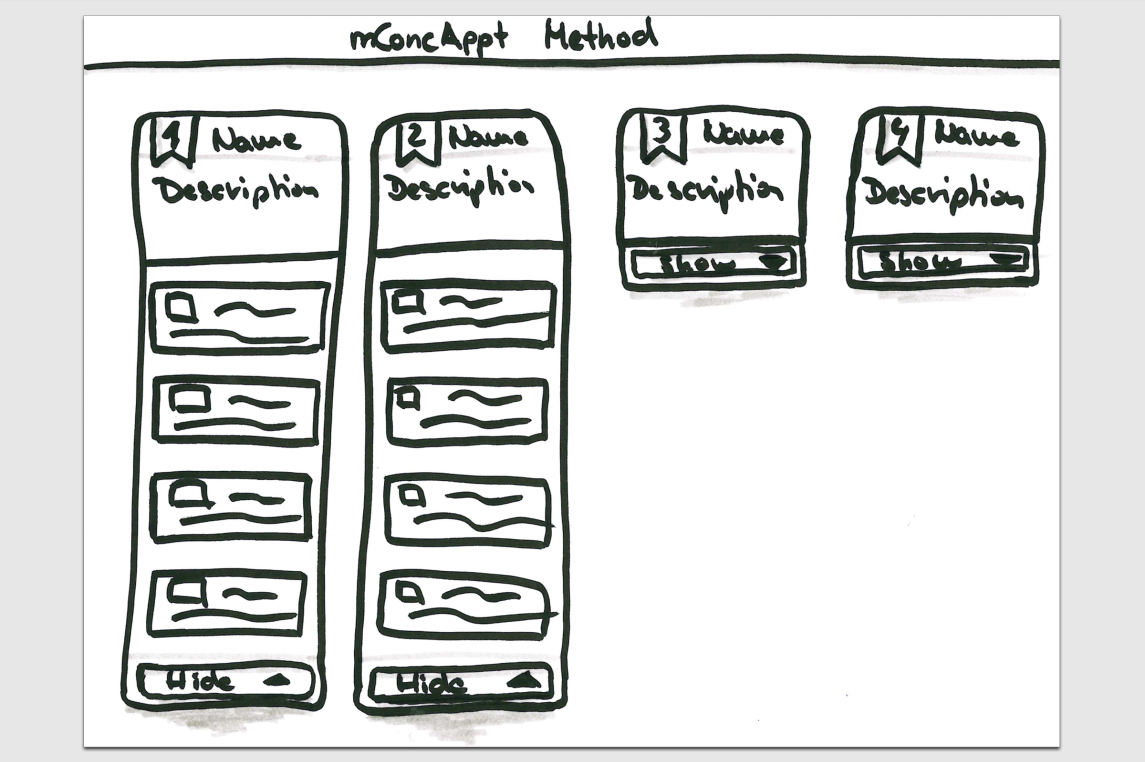


Figure 12: Fourth version indicates different states

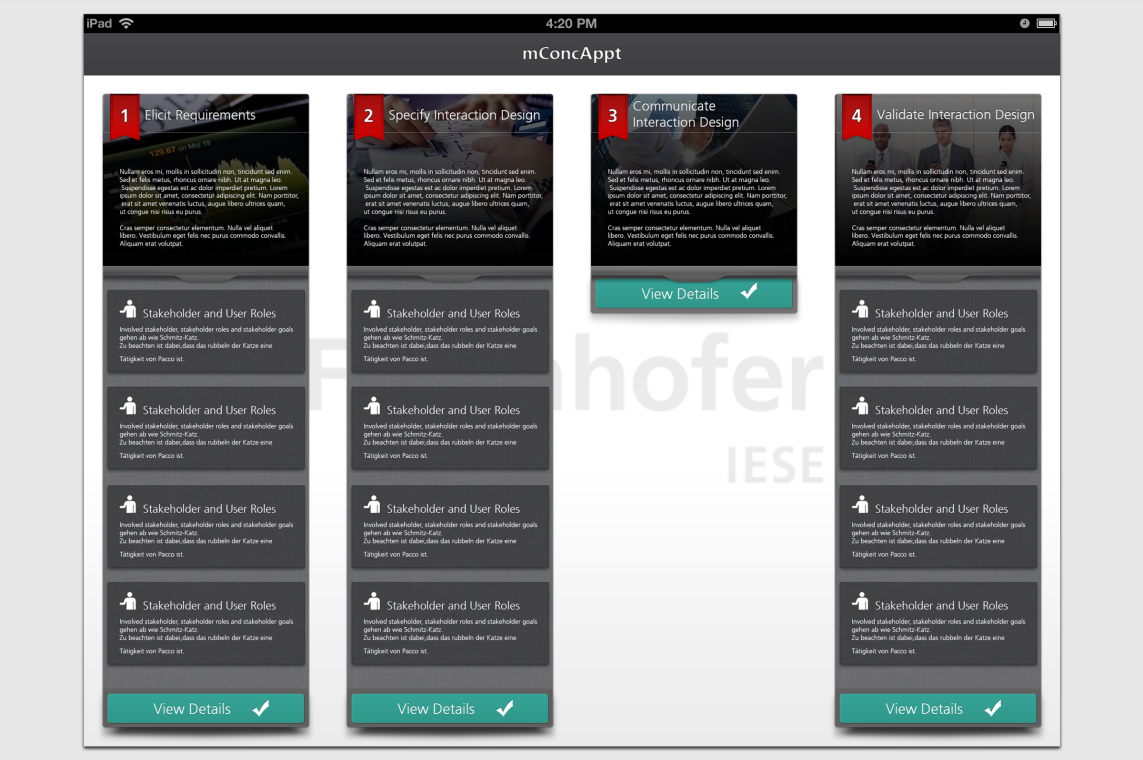


Figure 13: Almost final version of the screen

Rapid feedback on early wireframes can be given by colleagues or lead users. As shown in the introduction, a real user test is done after the first complete iteration of the method. Therefore a clickable prototype can easily be created using presentation software for non-linear presentations on the mobile device (e.g. Presentation Link).

## Model the Screen Flow (Task 2.5)

The *Screen Flow* is a holistic view on the complete navigation tree of the interaction concept. There should always be a flat navigation hierarchy without depth higher than 4. The user should always be able to quickly accomplish his tasks with the mobile system.

Figure 14 shows an example of a *Screen Flow*. Arrows show concrete navigation possibilities. Text at the arrows already gives a textual description of intended screen transitions. Screens are named with their unique identifier and a short title that allows the reader to comprehend the screens content without being able to match identifiers to concrete screens.

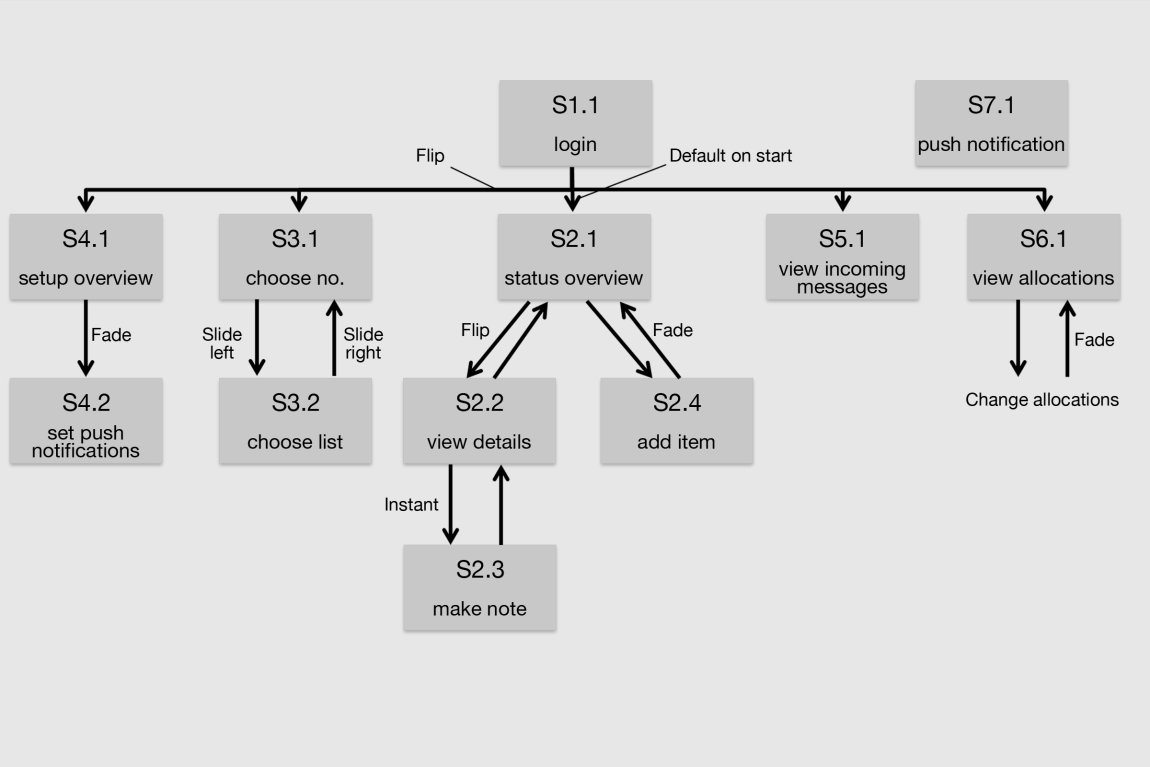


Figure 14: Screen flow example

# Phase 3: Validate Interaction Design

Within this phase, the following tasks are typically executed

|  |
| --- |
| * Task 3.1: Specify usage scenarios * Task 3.2: Create a clickable prototype * Task 3.3: Conduct user review |

Based on the interaction cases, concrete usage scenarios form the basis for conducting the test. In addition, a clickable prototype can be created easily using presentation software for non-linear presentations on the mobile device (e.g., Presentation Link). This approach requires marginally more time than applying Wizard-of-Oz testing with paper prototypes, but enables us to perform a user review on the actual end device in the concrete usage environment, which we think is mandatory especially for mobile business apps.

To support these tasks, you can find further information in form of presentations.

1. This guideline is an extract of the publication “S. Hess, F. Kiefer, R. Carbon, A. Maier, “mConcAppt - a method for the conception of mobile business applications”, Reportnr. 098.12/E ; http://publica.fraunhofer.de/documents/N-225635.html [↑](#footnote-ref-1)
2. Hippel, E. (1986), "Lead Users: A Source of Novel Product Concepts", Management Science 32(7): 791–806. [↑](#footnote-ref-2)
3. See <http://keynotopia.com/themes/> [↑](#footnote-ref-3)