# Design Documentation

# Free choice

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| 22/09/21 | 1.0 | Initial discussion of design | Fanny Söderlund, Malek Alabed, Nishat Jahan, Suzanne Zomer |
| 06/10/21 | 1.1 | Design of mock lab | Fanny Söderlund, Malek Alabed, Nishat Jahan, Suzanne Zomer |
| 20/10/21 | 1.2 | Additions of figures 4 and 5 | Malek Alabed |
| 21/10/21 | 1.3 | Addition of D4 and complementing figure 4 | Fanny Söderlund |
| 23/10/21 | 1.4 | Rearranging text, check spelling and details | Fanny Söderlund, Malek Alabed, Nishat Jahan, Suzanne Zomer, Ismail Eyamba |
| 23/10/21 | 1.5 | Addition of explanation of ‘mocked environment’ in D3 | Suzanne Zomer |
| 23/10/2021 | 1.6 | Additions of figure 2 and 3 | Nishat Jahan |

Design item List

|  |  |  |
| --- | --- | --- |
| **Design name** | **Requirements related** | **Priority** |
| D1. Client android app environment | R1, R2, R3 | Essential |
| D2. Client web app environment | R1, R2, R3 | Essential |
| D3. Server/API connection | R2, R4, R5, R6, R8 | Essential |
| D4. Home page designs | R4, R5, R7, R8, R9 | Desirable |

Design Item Descriptions

**D1.**

A collection of the designs for the requirements relating to the android client side of the system.

***Mocked environment***

The mocked environment allows the features to be developed incapsulated without outside interaction. The mocked client app is a simple android app that contains the necessary components for the related requirements. The mocked environment is created using Android Studio using Java to integrate with the other subgroups. As seen in *figure 1,* the mocked environment aims to be simple with a few components to make it easy for us to develop our features in it.

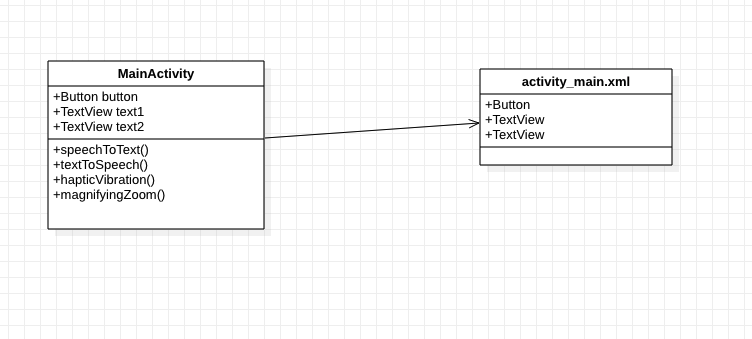
******

Figure 1 - Class diagram of mock client android app

Diagram, engineering drawing

Description automatically generated

**Figure 2: use case diagram 1 Mock client android app**

**Diagram

Description automatically generated**

**Figure 3: use case diagram 2 Mock client android app**

**D2.**

A collection of the designs for the requirements relating to the web client side of the system.

***Mocked environment***

The mocked environment allows the features to be developed incapsulated without outside interaction. The mocked client app is a simple web app that contains the necessary components for the related requirements. *Figure 2* shows the shell of the mock environment of the web client. *Figure 3* shows the connection of the mocked web client to the sever, whether a mocked API call or the actual system server.

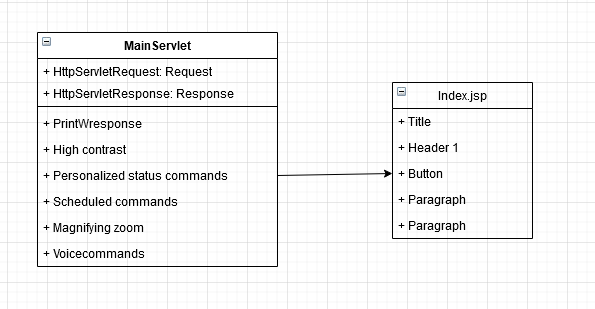


Figure 4- Class diagram of mock client android app

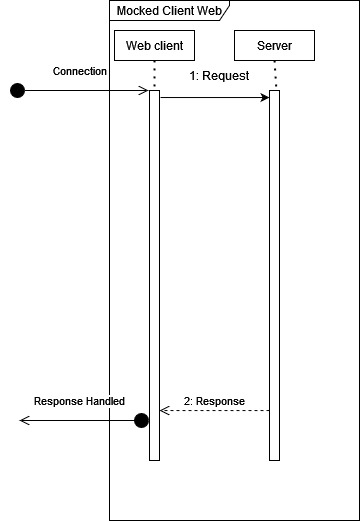


Figure 5 - Sequence diagram of connection of the mocked web client

**D3.**

A collection of the designs for the requirements relating to the server of the system. Initially, for testing and developing, the mocked environment will be used, but eventually the implementations will migrate to the actual server/API.

***Mocked environment***

The mocked environment allows the features to be developed incapsulated without outside interaction. The mocked server allows for testing and implementing of features not yet handled by the current system.

The platform used to create a mocked API server is [Postman](http://www.postman.com). It is commonly used for building or working with API’s. The platform allows for making HTTP requests to public and private servers and provides the response from the corresponding server. This will be beneficial for testing API calls to our own server throughout development of the project. Not only does the platform allow us to work as a team in the same workspace, but it also let the team manage the progress and keeps track of the history.

Postman has a feature to create your own mock API server. With this HTTP requests can be made with a simulated server response. The platform generates a URL with customized paths to make requests from the client side, to request and receive data in JSON format from/to the server. This appears useful for the early stages of our project to develop and implement new features for the Android application and web app.

******

Figure 4 - General layout of mocked sever/client interaction

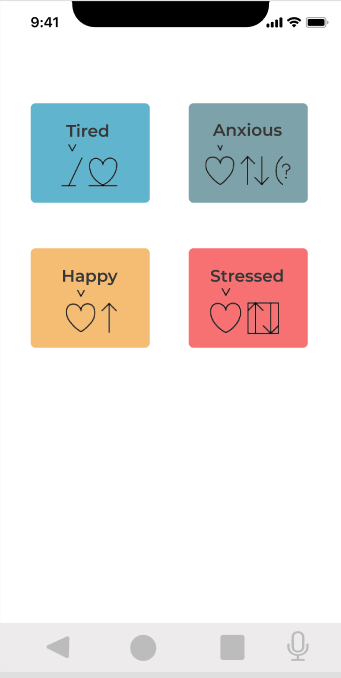
**D4.**

A collection of the designs for the requirements related to the home page, meaning all requirements that are features that the user does not toggle on and off. These features are constant to the system much like the feature to turn on and off different devices.

***R4 – Personalized status commands***

This feature is a home page where the user can set different command buttons that perform different actions set by the user. More is discussed in our requirements document under [R4].

For designing and implementing this feature, following figures (Figure 3) have been made. This is not a definite look of the system, but it gives an understanding of the requirement itself and allows us as a group to explain it better to other subgroups and be clear within the group how the feature should function.

 En bild som visar text, iPod, elektronik, skärmbild



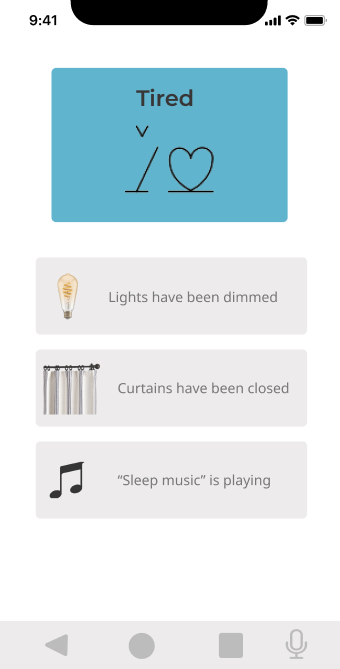
Automatiskt genererad beskrivning 

Figure 5 - Explainational figures of R4 (Figma, n.d.)

# Contents

Figma. (u.d.). *Figma* . Hämtat från Figma presentations: https://www.figma.com/proto/qpTRnpeAb3j8zieV8muJEV/Mood-commands?node-id=3%3A360&scaling=min-zoom&page-id=0%3A1