Change on project scope

From the requirements and specification document to the design document and the iteration one, the security system is changed from PIN code to login system. There is no change on project scope from the design document to the iteration one.

Change on iteration plan

Most backend functionality implementations are moved back to iteration two. The major reason was that we underestimate time units and difficulty of connecting servers with clients and setting the main server database. The changed implementation plan is listed below.

# Implementation Plan

**Legend:**

Anna: Project Manager

Ali/Joon: Front-End Developers

Andy/Na: Back-End Developers

Jack: Software Architect

**Iteration 0:**

Pre-programming to setup version control.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Task | Priority (1-5) | Difficulty (1-5) | Time Units (1-5) | Dependency | Person Responsible |
| Setup GitHub remote repository | 1 | 1 | 1 | None | Anna Kim |
| Fork Repositories | 1 | 1 | 1 | None | Andy, Joon, Ali, Na |
| Implement skeleton classes for all the activities | 1 | 1 | 1 | None | Joon/Ali/Jack |

**Iteration 1:**

Iteration one will focus on setting the foundation for our app by setting up things like the database (MySQL and java request server), logging into the app successfully and setting up the home screen. Most of the front-end screens will exist by the end of iteration 1 and the backend will start to be implemented as well. The connector class, the class that is used to fetch information from the back-end and display it on the front-end, will also be implemented by the end of iteration 1.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Task | Priority (1-5) | Difficulty (1-5) | Time Units (1-5) | Dependency | Person Responsible |
| Implement the database and java request server | 1 | 4 | 5 | None | Andy/Na |
| Implement Front End for “Create User” page, “Login” page, “Home” page, “Settings” page, “Goals” page, and “Categories” page, “Report” page, and “Calculator” page | 1 | 1 | 2 | None | Joon/Ali/Anna/Jack |
| Implement the functionality of all the MySQL and client/server methods so that they properly interact with the rest of the codebase | 5 | 4 | 3 | None | Jack |
| Confirm/Test with other developers that the database methods work properly in the code they are writing | 3 | 2 | 2 | Methods will have to be developed | Jack/Anna |
| Update requirements and specification document and design document | 3 | 1 | 3 | While or after coding, update them | All |
| Progress report | 2 | 1 | 2 | After completing rest of iteration 1 tasks | All |

**Iteration 2:**

Iteration 2 will focus on finalizing setting up queries and request handlers for the server as well as adding handlers for buttons and linking the front-end pages to the back-end data. User functionality will come together here, as by the end of the iteration, they will be fully linked to begin testing.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Task | Priority (1-5) | Difficulty (1-5) | Time Units (1-5) | Dependency | Person Responsible |
| Implement Back End for “Create User”, “Login”, “Report”, “Goals”, and “Category” pages so ensure that they link with the front-end properly and store the correct data. | 3 | 4 | 2 | Implementing Front End for Pages | Andy/Na/Anna |
| Add Handling for invalid usernames and passwords and if profile already exists as well as changing passwords | 1 | 1 | 1 | Implementing Back end of the login page | Andy/Na |
| Add a Handler for a “Create Account” Button | 1 | 1 | 1 | Will need Create Account Button to be developed | Joon/Ali |
| Implement Back end for calculating current expense, current balance, getting categories, adding transaction, getting daily graph. | 1 | 2 | 2 | None | Andy/Na |
| Add Back end Logic for editing a users profile, sending notifications, user guide, getting application file | 1 | 2 | 4 | None | Andy/Na |
| Implement the functionality of the connecting class between the front-end and the back-end so that they properly interact with the rest of the code | 1 | 4 | 3 | Will need back-end data along with front-end data handlers to ensure this is done properly | Jack/Na |
| Add handlers to link the front-end and back-end for category and goals page | 1 | 1 | 1 | Both goals and categories front-end and back-end must exist | Joon/Ali/Anna |
| Implement handler to add and edit the categories | 1 | 1 | 1 | Edit categories button must already exist | Joon/Ali |
| Implement table using the info from the database to be able to display the categories and the user’s income | 1 | 2 | 2 | Categories page must already exist and data must be already stored in the back-end | Andy/Na/Anna |
| Add post account creation information filling, and notification for user to go to user guide | 1 | 2 | 2 | Account must already be created and stored in the back-end along with user data | Ali/Joon |
| Integrate database with logging in backend logic | 3 | 3 | 2 | Database must be defined | Andy/Na |
| Implement Database management code for deleting older data for both MySQL and the server | 3 | 2 | 2 | Data must already be stored in the database to do this. | Jack/Anna |

**Iteration 3:**

Iteration 3 will focus on testing our app to make sure everything has come together as planned. We will need to make sure that the front end of the UI comes together and is able to fetch data from the back-end as needed.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Task | Priority (1-5) | Difficulty (1-5) | Time Units (1-5) | Dependency | Person Responsible |
| Implement Front-End of the Calculator tab if time permits | 1 | 3 | 3 | None | Ali/Joon |
| Implement Back-End of the Calculator tab if time permits | 1 | 3 | 3 | None | Andy/Na |
| Testing to make sure the Front-End is getting the correct data from the back-end | 5 | 5 | 5 | Everything in the front-end and back-end must be implemented | Jack/Anna |
| Test to make sure everything in the front-end works as it is supposed to | 5 | 5 | 5 | Everything in the front-end must be implemented | Ali/Joon/Anna |
| Test to make sure everything in the back-end is both calculated correctly as well as stored correctly | 5 | 5 | 5 | Everything in the back-end must be implemented | Na/Andy/Anna |