



Requirements document

Project course in software development



*Authors: Student 1,
Student 2,
Student 3,
Student 4,
Student 5
Semester: VT20
Course code: 2DV609*



Table of Contents

Table of Contents	2
Requirements document	4
Introduction	4
Stakeholders	4
Requirement elicitation	5
Requirement analysis keywords	7
Checklist based requirement analysis	8
Analysis of requirements checklist	9
Changes to requirements	11
Classify requirements	13
Keywords	13
Requirement risks	15
Risk keywords	15
Overall risks	17
Validation of Requirements	18
Analysis of Validation Checklist	20
Final Requirements	21
Use Cases	24
UC1 - Getting Started	25
UC2 - Login	26
UC3 - Logout	27
UC4 - Create A New Habit	28
UC5 - Personal Details	29
Test Cases	31
Overview	31
Functional	33
SIGNUP - Sign up new User	33
LOGIN - Login Existing User	34
LOGOUT - Logout Existing User	34
CREATEHAB - Create Habit	35
CATEGHAB- Categorize Habit	36
ADDHAB - Add Habit to Category	37
REMINDER - Send Email Reminder	38
PROFILE - User Profile	39
EDITDETAILS - Edit Personal Details	40
VIEWDETAILS - View Personal Details	41
PROGRESS - Current Progress	41



DIFFICULTY - Habit Difficulty Setting	42
AUTH-USER - User Authentication	43
UNIQHAB - Unique Habit Creation	44
IDTAG - Unique User ID	44
Non-Functional	45
DATABASE - Database	45
C.RESPONSE - Client Login Response Time	46
S.RESPONSE - Server Login Response Time	46
MVCPATTERN - MVC Pattern	47
MULT.CATEG. - Multiple Categories	48
ADDCATEG. - Add Habit to Categories	49
EMAILREM - Email Reminder	50
Performance Requirement Modelling with MARTE	51
Activity diagram	51
Model UML Requirements	52
Class Diagrams	52
LOGIN	52
SIGNUP	53
LOGOUT	54
Habit	55
ADDHAB	56
REMIND	56
CREATHAB	57
EDITPD	57
VIEWPD	58
PROGRESS	58
USERPRO	59
HABDIFF	60
FIREBASE	60
MULT.CATEG	61
EMAILREM	61
State/Activity Diagrams	62
C.RESPONSE	62
S.RESPONSE	62
ID-TAG	63



Requirements document

Introduction

Our team has gotten a job request to create an application for a company.

The goal of our application is to motivate users to create new habits and keep their habits over time.

To motivate the users we will use a reward system and help keep their habits we will use a scheduling system and remind the users if they miss their habit timings. To use the application the users have to create an account that they can log into. The application will be web based and will be coded with Typescript, Javascript and the library React.

This document will specify all the requirements we need in the application and the requirements will go through analysis and will be changed if they deem unfinished. Then we will have test cases for the requirements and finally some UML that will visualize the requirements.

Stakeholders

End-user

Project developers

Project managers

Project owners

User

Requirement elicitation

In this section, we initialize the requirements, requirement identifier, and type. The initial requirement will most likely change during the document. The requirement identifier will be used instead of the full requirements in future sections.

Requirement	Requirement identifier	Type
A user shall be able to log in.	LOGIN	Functional
A user shall be able to sign up.	SIGNUP	Functional
A user shall be able to log out.	LOGOUT	Functional
A user shall be able to create an object that describes a repetitive behavior.	HABIT	Functional
Users shall be able to create unique habits by describing a habit and time to do the habit.	UNIQHABIT	Functional
A habit shall consist of a title, description, which category the habit belongs to, start time and end time, how frequent the habit shall occur and a way to notice if a habit is not completed.	HABCONTENT	Functional
A user shall be able to categorize their habits.	CATHAB	Functional
A user shall be able to create a habit.	CREATHAB	Functional
A user shall be able to add a habit to a category.	ADDHAB	Functional
A user should be able to see their progression.	PROGRESS	Functional
A user shall get a reminder through email about their habits.	REMIND	Functional
A user shall have a unique id tag.	IDTAG	Functional
A user shall be the only one that can access their personal profile.	AUTH-USER	Functional
A user shall have a user profile.	USERPROFILE	Functional



A user shall be able to edit their personal information.	EDITPI	Functional
A user shall be able to see their personal information in the user profile.	SEEIP	Functional
A user can specify the difficulty of the habit.	HABDIFF	Functional
The login shall take less than 10 seconds under optimal circumstances.	LOGIN2	Non-functional
The application shall be based on the Flux pattern.	FLUX	Non-functional
The application shall have more than two categories for habits.	MULTCAT	Non-functional
A user shall be able to add a habit to one or multiple categories.	ADDMULTCAT	Non-functional
The application shall send an email at most 5 minutes after failing to do a session of habit.	EMAILREM	Non-functional



Requirement analysis keywords

In this section, we will specify what the questions that will be used in the analysis mean.

Premature design	Does the requirement include the premature design or implementation information?
Combined requirements	Could the description of a requirement be broken down into several different requirements?
Unnecessary requirements	Is the requirement 'gold-plating'? That is a cosmetic addition to the system which is not really necessary.
Use of non-standard hardware	Does the requirement mean that non-standard hardware or software must be used?
Conformance with business goals	Is the requirement consistent with the business goals defined in the introduction to the requirements document?
Requirements ambiguity	Is the requirement ambiguous, i.e. could it be read in different ways by different people?
Requirements realism	Is the requirement realistic given the technology which will be used to implement the system?
Requirements testability	Is the requirement testable, that is, is it stated in such a way that test engineers can derive a test that can show if the system meets that requirement?



Checklist based requirement analysis

This section will include a checklist to analyze the requirements, the checklist questions will be the one from the PowerPoint RE1 and have been further explained in the previous section. Depending on if the requirement fulfills the questions or not the checklist will be filled with YES or NO. After the checklist, there will be an analysis of the checklist result.

Req. Identifier	Premature design	Combined Req.	Unnecessary Req.	Use of non-standard hardware	Conformance with business goals	Req. ambiguity	Req. realism	Req. testability
LOGIN	NO	NO	NO	NO	YES	NO	YES	YES
SIGNUP	NO	NO	NO	NO	YES	NO	YES	YES
LOGOUT	NO	NO	NO	NO	YES	NO	YES	YES
HABIT	NO	NO	NO	NO	YES	NO	YES	YES
UNIQHABIT	NO	NO	NO	NO	YES	NO	YES	YES
HABCONTENT	NO	YES	NO	NO	YS	YES	YES	YES
CATHAB	NO	NO	NO	NO	YES	YES	YES	YES
CREATHAB	NO	NO	NO	NO	YES	NO	YES	YES
ADDHAB	NO	NO	NO	NO	YES	NO	YES	YES
PROGRESS	NO	NO	YES	NO	YES	YES	YES	YES
REMIND	NO	NO	NO	YES	YES	YES	YES	YES
IDTAG	NO	NO	NO	NO	YES	NO	YES	YES
AUTH-USER	NO	NO	NO	NO	YES	NO	YES	YES
EXCRYPTION	NO	NO	NO	YES	YES	NO	YSE	YES
USERPROFILE	NO	NO	NO	NO	YES	NO	YES	YES
EDITPI	NO	NO	NO	NO	YES	YES	YES	YES
SEEPI	NO	NO	NO	NO	NO	YES	YES	YES
HABDIFF	NO	NO	YES	NO	NO	NO	YES	YES
C.RESPONSE	NO	YES	NO	NO	NO	NO	YES	YES
S.RESPONSE	NO	YES	NO	NO	NO	NO	YES	YES
MVCPATT	YES	NO	NO	NO	NO	NO	YES	NO
MULTCAT	NO	NO	NO	NO	NO	NO	YES	YES
ADDMULCAT	NO	NO	NO	NO	NO	NO	YES	YES
EMAILREM	NO	NO	NO	YES	NO	NO	YES	YES



Analysis of requirements checklist

In the checklist, we can see that there are quite a few requirements that have some issues, after analyzing the result some requirements will change.

First, we have HABCONTENT which is a combined requirement, it is quite clear that the requirement could be divided into sections for each of the values inside a habit. That will be the solution to fix the current issue with a combined requirement. The values are also a little bit ambiguous which will be solved with having a requirement for each of the values.

Then, we have CATHAB where it is a bit unclear for example what does categorize habits mean. It could be rephrased where it is made clear that habits can be put into categories. After that we have the requirement PROGRESS which is a bit unnecessary and unclear, the reason why it could be seen as unnecessary is the fact that the requirement is not crucial for the application to work. It is good to have the requirement though since it will be used to motivate the users and is consistent with the business goals. The requirement will be refactored to be more specific and clear to try solving the current issue.

Then we have REMIND which is unclear since it specifies that the application will send an email to remind the user that they have a habit to do. The current requirement makes it hard to understand if the reminder is sent before a habit or after if they miss it. This shows that there is a need to rephrase the requirement to make it more clear when the notification should be sent. REMIND will also use non-standard software from a third party, we don't know a way to solve this requirement without using third-party software.

In the EDITPI requirement, it is a bit unclear since what is included in personal information, but at the same time, it is hard to specify details without making it too big of a requirement so at this point it just needs to be rephrased. SEEPI has a similar issue as EDITPI since personal



information isn't specified, but as written before it should just be rephrased, but not change the content inside it.

After that we have MVCPATT, MVC is a design pattern we are planning to use and it specifies how we are planning to implement the application. It's still a necessary requirement to include since it will help to structure the application which will make more parts of the code reusable and flexible.

The identifier HABDIFF will be changed to DIFFICULTY since it is deemed more clear what it requirement it belongs to.

The final change to the requirements will be to LOGIN2 since it is seen as a combined requirement, it will be split into two requirements C.RESPONSE and S.RESPONSE.

There are also multiple requirements that are not consistent with the business goals, but they are still necessary for the application to work, so they don't need to be changed.



Changes to requirements

The requirement that will be changed will be stated first and then below the new version will be specified.

HABONTENT - A habit shall consist of a title, description, which category the habit belongs to, start time and end time, how frequent the habit shall occur and a way to notice if a habit is not completed.

DATE - An object shall contain yyyy/mm/dd - HH/MM/ss.

TITLE - An object shall contain a title.

DESCRIPTION - An object shall contain a short description.

FREQUENCY - An object shall contain how frequent an event shall occur.

NOTIFIER - An object shall contain a check if the event has been completed.

CATHAB - A user should be able to categorize their habits.

HABCAT - A user should be able to put habits in categories.

PROGRESS - A user should be able to see their progression.

PROGRESS - A user shall be able to see their weekly progression based on their performance when completing habits.

REMIND - A user should get a reminder through email about their habits.

REMIND - A user should get a reminder through email if they miss completing a habit.

EDITPI - A user should be able to edit their personal information.

EDITPD - A user should be able to edit their personal details.

SEEPI - A user should be able to see their personal information in the user profile.

VIEWPD - A user should be able to view their personal details.

MVCPATT will be changed to **MVCPATTERN** to make the identifier more clear.

HABDIFF identifier will be changed to **Difficulty** identifier.



LOGIN2 - The login shall take less than 10 seconds under optimal circumstances.

C.RESPONSE - Client side login should take less than 10 seconds under optimal circumstances.

S.RESPONSE - Server Side login should take less than 5 seconds under optimal circumstances.



Classify requirements

In this section, there will be a classifier of the requirements we have chosen. To do so we will use a set of keywords to classify each requirement and check if the keyword matches the requirement or not.

Keywords

System
<ul style="list-style-type: none">• Requirements that affect the entire system such as performance or reliability requirements.
User interface
<ul style="list-style-type: none">• Requirements that are concerned with user interaction.
Database
<ul style="list-style-type: none">• Requirements that are concerned with the data managed by the system.
Communications
<ul style="list-style-type: none">• Requirements that are concerned with the external communication facilities in the system.
Security
<ul style="list-style-type: none">• Requirements that are concerned with the security of the application



Req Identifier	System	User Interface	Database	Communications	Security
LOGIN	YES	YES	YES	NO	YES
SIGNUP	YES	YES	YES	NO	YES
LOGOUT	YES	YES	YES	NO	NO
HABIT	YES	YES	YES	NO	NO
UNIQHABIT	NO	YES	YES	NO	NO
CATHAB	YES	YES	YES	NO	NO
CREATHAB	YES	YES	YES	NO	NO
ADDHAB	YES	YES	YES	NO	NO
PROGRESS	YES	YES	YES	NO	NO
REMIND	YES	NO	NO	YES	NO
IDAG	NO	NO	NO	NO	YES
AUTH-USER	YES	YES	YES	NO	YES
USERPROFILE	NO	NO	YES	NO	YES
EDITPI	YES	YES	YES	NO	YES
SEEPI	NO	YES	YES	NO	YES
HABDIFF	NO	YES	NO	NO	NO
C.RESPONSE	YES	NO	YES	NO	NO
S.RESPONSE	YES	NO	YES	NO	NO
MVCPATTERN	NO	NO	NO	NO	NO
MULTCAT	NO	YES	YES	NO	NO
ADDMULTCAT	NO	NO	YES	NO	NO
EMAILREM	YES	NO	NO	NO	NO
DATE	NO	YES	YES	NO	NO
TITLE	NO	YES	YES	NO	NO
DESCRIPTION	NO	YES	YES	NO	NO
FREQUENCY	NO	YES	YES	NO	NO
NOTIFIER	NO	YES	YES	NO	NO



Requirement risks

In this section, there will be a classifier of the requirement risks. To do so we will use a set of keywords to classify each requirement and check if the keyword matches the requirement or not. We will use a fuzzy approach and categories each risks with low, medium and high.

Risk keywords

Performance risks
<ul style="list-style-type: none">• May affect the overall performance of the system.
Safety and security risks
<ul style="list-style-type: none">• May cause problems in meeting overall system requirements for safety and security.
Process risks
<ul style="list-style-type: none">• May require changes to the normal development process, e.g., the introduction of mathematical specifications (for safety requirement) or the use of unfamiliar prototyping systems (for UI requirement).
Implementation technology risks
<ul style="list-style-type: none">• May require the use of unfamiliar implementation technology, such as AI techniques, the use of N-version programming for fault tolerance, etc.
Database risks
<ul style="list-style-type: none">• May involve non-standard data that is not available in an existing system database.
Schedule risks
<ul style="list-style-type: none">• May be technically difficult and may threaten the planned development schedule for the system.
External risks
<ul style="list-style-type: none">• Involves external contractors.
Stability risks
<ul style="list-style-type: none">• Requirements may be volatile and subject to evolution during the development process.



Req. Identifier	Performance	Safety and security	Process	Implementation technology	Database	Schedule	External	Stability
LOGIN	Medium	Medium	Low	Low	Low	Low	Low	Medium
SIGNUP	Medium	Medium	Low	Low	Low	Low	Low	Medium
LOGOUT	Medium	Medium	Low	Low	Low	Low	Low	Medium
HABIT	Low	Low	Low	Medium	Medium	Medium	Low	Low
UNIQHABIT	Low	Low	Medium	Medium	Low	Low	Low	Low
CATHHAB	Low	Low	Low	Low	Low	Low	Low	Low
CREATHAB	Low	Low	Low	Low	Low	Low	Low	Low
ADDHAB	Medium	Low	Low	Low	Low	Low	Low	Low
PROGRESS	Medium	Low	Medium	Medium	Low	Low	Low	Medium
REMIND	Medium	Low	Low	Medium	Low	Medium	Low	Medium
IDTAG	Low	Low	Low	Low	Medium	Low	Low	Low
AUTH-USER	Low	Medium	Low	Medium	Medium	Low	Low	Medium
USERPROFILE	Medium	Medium	Low	Low	Low	Medium	Low	Low
EDITPI	Medium	Medium	Low	Low	Low	Low	Low	Medium
SEETPI	Low	Low	Low	Low	Low	Low	Low	Low
HABDIFF	Low	Low	Medium	Medium	Low	Medium	Low	Low
C.RESPONSE	High	Medium	Low	Low	Low	Low	Low	Medium
S.RESPONSE	High	Medium	Low	Low	Low	Low	Low	Medium
MVCPATTERN	Low	Low	Low	Low	Low	Low	Low	Low
MULTCAT	Low	Low	Medium	Medium	Low	Low	Low	Low
ADDMULTCAT	Low	Low	Medium	Medium	Low	Low	Low	Low
EMAILREM	Medium	Low	Low	Medium	Low	Medium	Low	Medium
DATE	Low	Low	Low	Low	Low	Low	Low	Low
TITLE	Low	Low	Low	Low	Low	Low	Low	Low
DESCRIPTION	Low	Low	Low	Low	Low	Low	Low	Low
FREQUENCY	Low	Low	Medium	Medium	Medium	Medium	Low	Low
NOTIFIER	Low	Low	Medium	Medium	Low	Low	Low	Low



Overall risks

In this section, we show the overall risk of each requirement by taking the average risk from the requirement risk table.

Requirement identifier	Average risk
SIGNUP	Medium
LOGIN	Medium
LOGOUT	Medium
CREATHAB	Low
HABCAT	Low
ADDHAB	Low
REMIND	Medium
USERPRO	Medium
EDITPD	Medium
VIEWPD	Low
PROGRESS	Medium
CATEG.HAB	Medium
S.RESPONSE	Medium
C.RESPONSE	Medium
MVCPATTERN	Low
MULTCAT	Low
ADDMULTCAT	Low
EMAILREM	Medium
IDTAG	Low
AUTH-USER	Medium
HABIT	Low
UNIQHABIT	Low
DATE	Low
TITLE	Low
DESCRIPTION	Low
NOTIFIER	Low
FREQUENCY	Medium



Validation of Requirements

In this section the requirements are analysed and evaluated. The questions are based on usability of and their performance.

Identifier	Are the requirements complete?	Are the requirements consistent?	Are the requirements comprehensible?	Are the requirements ambiguous?	Is the requirements document structured?	Are the requirements traceable?
LOGIN	Y	Y	Y	N	Y	Y
SIGNUP	Y	Y	Y	N	Y	Y
LOGOUT	Y	Y	Y	N	Y	Y
HABCAT	Y	Y	Y	N	Y	Y
CREATHAB	Y	Y	Y	N	Y	Y
ADDHAB	Y	Y	Y	Y	Y	Y
PROGRESS	Y	Y	Y	N	Y	Y
REMIND	Y	Y	Y	N	Y	Y
USERPRO	Y	Y	Y	N	Y	Y
EDITPD	Y	Y	Y	Y	Y	Y
VIEWPD	Y	Y	Y	N	Y	Y
DIFFICULTY	Y	Y	Y	N	Y	Y
C.RESPONSE	Y	Y	Y	N	Y	Y
S.RESPONSE	Y	Y	Y	N	Y	Y
MVCPATTERN	Y	Y	Y	N	Y	Y
MULTCAT	Y	Y	Y	N	Y	Y
ADDMULCAT	Y	Y	Y	N	Y	Y
EMAILREM	Y	Y	Y	N	Y	Y
IDTAG	Y	Y	Y	N	Y	Y
AUTH-USER	Y	Y	Y	N	Y	Y
HABIT	Y	Y	Y	N	Y	Y
UNIQHABIT	Y	Y	Y	N	Y	Y
DATE	Y	Y	Y	N	Y	Y



TITLE	Y	Y	Y	N	Y	Y
DESCRIPTION	Y	Y	Y	N	Y	Y
FREQUENCY	Y	Y	Y	N	Y	Y
NOTIFIER	Y	Y	Y	N	Y	Y
PROGRESS	Y	Y	Y	N	Y	Y



Analysis of Validation Checklist

The identifiers are placed correctly, corresponding to their specified requirement. The checklist shows signs of explicit perspective on requirements, however, there are some issues with some requirements ambiguity. Such as ADDHAB, which states that a user should be able to categorize their habits. Does this requirement bring questions such as: are there any predefined categories? The same ambiguity goes for EDITPI. It brings up the question: which part of their profile a user should a user be able to edit. When a user wants to categorise their habits, it brings up the question to mind: how should a user approach this task? HABCAT was ambiguous regarding that requirement, later it specified that a user should be able to create a category for their habits. Besides those issues, all the requirements are traceable, consistent and structured.



Final Requirements

In this section, we define the final requirements after the previous sections where we explored the current requirements and sorted which are relevant to our software implementation.

Functional							
	Initial		Final				
	Identifier	Requirements	Identifier	Requirements	Use Case	Test Case	Type
1	SIGNUP	A user should be able to sign up.	SIGNUP	A user should be able to sign up.	UC1 - Get Started	Sign up new user Test	M
2	LOGIN	A user should be able to login.	LOGIN	A user should be able to login.	UC2 - Login	Login existing user Test	M
3	LOGOUT	A user should be able to logout.	LOGOUT	A user should be able to logout.	UC3 - Logout	Logout existing user Test	M
4	CREATHAB	A user should be able to create a habit.	CREATEHAB	A user should be able to create a habit.	UC4 - Create a new Habit	Create Habit Test	M
5	HABCAT	A user should be able to put habits in categories.	CATEGHAB	A user should be able to create categories for their habits.	-	Create Category Test	M
6	ADDHAB	A user should be able to add a habit to a category.	ADDHAB	A user should be able to add a habit to a category.	-	Add Habit to Category Test	M
7	REMIND	A user should get a reminder through email if they miss to complete a habit.	REMIND	A user should get a reminder through email if they miss to complete a habit.	-	Send Email Reminder Test	M
8	USERPRO	A user has a personal profile.	PROFILE	A user has a personal profile page.	-	User Profile Test	M
9	EDITPD	A user should be able to edit their personal details.	EDITPD	A user should be able to edit their personal details.	UC5 - Personal Details	Edit Personal Details Test	M



10	VIEWPD	A user should be able to view their personal details.	VIEWPD	A user should be able to view their personal details.	UC5 - Personal Details	View Personal Details Test	M
11	PROGRESS	A user should be able to see their progression through a progression bar after completing habits.	PROGRESS	A user should be able to see their progression through a progression bar after completing habits.	-	Current Progress Test	M
12	HABDIFF	A user can specify the difficulty of the habit. (If we have time)	DIFFICULTY	A user can specify the difficulty of the habit. (If we have time)	-	Habit Difficulty Setting Test	M
13	AUTH-USER	A user shall be the only one that can access their personal profile.	AUTH-USER	A user shall be the only one that can access their personal profile.	-	Auth-User Test	M
14	UNIQHAB	Users shall be able to create unique habits by describing a habit and time to do the habit.	UNIQHAB	Users shall be able to create unique habits by describing a habit and time to do the habit.	-	Unique Habit Test	M
15	IDTAG	A user shall have a unique id tag.	IDTAG	A user shall have a unique id tag.	-	ID tag Test	M
Non-Functional							
	Initial		Final				
	Identifier	Requirements	Identifier	Requirements	Use Case	Test Case	Type
1	FIREBASE	The applications database is firebase.	DATABASE	The applications database is firebase.	-	Database Test	M
2	LOGIN2	The login should take less than 10 seconds under	C.RESPONSE	Client side login should take less	-	Client Login Response Time	M



		optimal circumstances.		than 10 seconds under optimal circumstances.		Test	
3			S.RESPONSE	Server Side login should take less than 5 seconds under optimal circumstances.	-	Server Login Response Time Test	M
4	FLUX	The application should be based on the Flux pattern.	FLUX	The application should be based on the Flux pattern.	-	MVC Pattern Test	M
5	MULTCAT	The application should have more than two categories for habits.	MULTCATEG	The application should have more than two categories for habits.	-	Multiple Categories Test	M
6	ADDONECAT	The application should allow a habit to be added to only one category.	ADDCATEG	The application should allow a habit to be added to only one category.	-	Add Habit to Category Test	M
7	EMAILREM	The application should send a email at most 5 minutes after failing to do a habit.	EMAILREM	The application should send an email at most 5 minutes after failing to do a habit.	-	Email Reminder Test	M



Use Cases

In this section, we will list a few use cases of the system. We use use cases to analyse, identify, clarify and organize system requirements specified in the design process. A use case contains a set of possible sequences of interactions between systems and users in a particular environment and related to a specific goal. Use cases describe the functional requirements of a system from the end user's perspective, creating a goal-focused sequence of events that is easy for users and developers to follow.

The use cases shall contain the following characteristics:

- Organized functional requirements.
- Model the goals of the system and, or actor interactions.
- Scenarios from start events to end.
- Describes one main flow of events and various alternate flows.



UC1 - Getting Started

UC1 - Get Started	
Identifier	SIGNUP
Requirement	A user should be able to sign up.
Actor	End-User
Scenario	The form should accept the request for sign up and redirect the user to login / personal profile if automatic login is enabled.
Precondition	The website is open, the no user is currently signed in, and the user does not already have an account.
Steps	<ol style="list-style-type: none">1. Open Web Browser and enter 'WebServer.org'.2. Wait for the web page to load.3. Click 'Sign Up'.4. Fill in personal information:<ol style="list-style-type: none">a. First name enter 'John'.b. Last name enter 'Doe'.c. Age enter '36'.d. Sex enter 'Male'.e. Email enter 'johndoe@gmail.com'.f. Password enter '123veryGood'.g. Confirm Password enter '123veryGood'.5. Verify that all the information entered is correct.6. Click 'Continue'.7. Wait while WebServer redirects you to your Personal Profile.
Alternate Steps	-
Comments	Subject to change*



UC2 - Login

UC2 - Login	
Identifier	LOGIN
Requirement	A user should be able to login.
Actor	End-User
Scenario	The form should accept the request for login and redirect the user to their personal profile.
Precondition	The website is open, no user is currently signed in, and the user already has an account.
Steps	<ol style="list-style-type: none">1. Open Web Browser and enter 'WebServer.org'.2. Wait for the web page to load.3. Click 'Login'.4. Fill in login information:<ol style="list-style-type: none">a. Navigate to email field and enter 'johndoe@gmail.com'.b. Navigate to password field enter '123veryGood'.5. Click 'Continue'.6. Wait while WebServer redirects you to your Personal Profile.
Alternate Steps	-
Comments	Subject to change*



UC3 - Logout

UC3 - Logout	
Identifier	LOGOUT
Requirement	A user should be able to logout.
Actor	End-User
Scenario	The site should accept the request for logout and redirect the user to the home page.
Precondition	The website is open, the user is currently signed in.
Steps	<ol style="list-style-type: none">1. Navigate to the Profile Picture in the TOP-RIGHT corner.2. Click on the Profile Picture.3. Navigate down to Logout/Sign out.4. Click on 'Logout/Sign out'.5. Wait while WebServer redirects you to the Home Page.
Alternate Steps	-
Comments	Subject to change*



UC4 - Create A New Habit

UC4 - Create a new Habit	
Identifier	CREATEHAB
Requirement	A user should be able to create a habit.
Actor	End-User
Scenario	The user can create personal habits with a name describing the habit and with additional custom information.
Precondition	The website is open, the user is currently signed in and can view the 'My Habits' header of the profile page.
Steps	<ol style="list-style-type: none">1. (See UC4 - Alternate Steps 1a or 1b).2. Wait while the WebServer redirects you to the 'My Habits' page.3. Navigate to, and click 'Create new Habit'.4. The WebServer will internally load the fields for entering details about the new habit.5. Enter details:<ol style="list-style-type: none">a. Habit Name: 'Work on Project'.b. Category: 'Work'.c. Difficulty: 'Medium'.d. Frequency enter 'Every day'.e. Set Reminder enter '4:00 pm'.6. Click 'Finalize'.
Alternate Steps	<ol style="list-style-type: none">1a. From the 'Home' page navigate to 'My Profile' and click the header.2a. (See UC4 - Alternate Step 1b). <hr/> <ol style="list-style-type: none">1b. From the 'My Profile' page navigate to 'My Habits' and click the header.
Comments	Subject to change*



UC5 - Personal Details

UC5 - Personal Details	
Identifier	EDITPD & VIEWPD
Requirement	A user should be able to edit their personal details. & A user should be able to view their personal details.
Actor	End-User
Scenario	The user wishes to edit their personal information such as name, email etc. & The user wishes to view their personal information such as name, email etc.
Precondition	The user is logged in and currently in their Personal Profile.
Steps	<ol style="list-style-type: none">1. (See UC5 - Alternate Steps 1a or 1b).2. Wait while the WebServer redirects you to the 'My Information' page.3. View your personal information/details.4. Navigate to, and click 'Edit Details'.5. The WebServer will internally load the fields for editing your personal details.6. Edit details.7. Scroll down to the bottom of the Personal information/details page and click 'Change Password'.8. Edit Password:<ol style="list-style-type: none">a. Old Password: '123veryGood'.b. New Password: '123VeryGood'.c. Repeat New Password: '123VeryGood'.7. Click 'Finalize'.8. Wait while the WebServer authenticates your new Password and (See UC5 step 2).
Alternate	<ol style="list-style-type: none">1a. From the 'Home' page navigate to



Steps	'My Profile' and click the header. 2a. (See UC5 - Alternate Step 1b). ----- 1b. From the 'My Profile' page navigate to 'My Information' and click the header.
Comments	Subject to change*



Test Cases

In this section, we will list the Manual Test Cases for the WebServer. A Test Case has a set of test data as well as conditions set, for a Test Case to succeed it needs to correspond to the Expected Result/Output and Post-Conditions already set before testing. A Test Case is the first step in Testing and is used by the Tester as a Step-by-step guide on how to perform a certain task, so it can be replicated by other testers in the future if there are, for example, any bugs present.

Overview

Functional					
	Identifier	Requirements	Use Case	Test Case	Type
1	SIGNUP	A user should be able to sign up.	UC1 - Get Started	Sign up new user Test	M
2	LOGIN	A user should be able to login.	UC2 - Login	Login existing user Test	M
3	LOGOUT	A user should be able to logout.	UC3 - Logout	Logout existing user Test	M
4	CREATEHAB	A user should be able to create a habit.	UC4 - Create a new Habit	Create Habit Test	M
5	CATEGHAB	A user should be able to create categories for their habits.	-	Create Category Test	M
6	ADDHAB	A user should be able to add a habit to a category.	-	Add Habit to Category Test	M
7	REMIND	A user should get a reminder through email if they miss to complete a habit.	-	Send Email Reminder Test	M
8	PROFILE	A user has a personal profile page.	-	User Profile Test	M
9	EDITPD	A user should be able to edit their personal details.	UC5 - Personal Details	Edit Personal Details Test	M
10	VIEWPD	A user should be able to view their personal details.	UC5 - Personal Details	View Personal Details Test	M
11	PROGRESS	A user should be able to see their progression through a progression bar after completing habits.	-	Current Progress Test	M



12	DIFFICULTY	A user can specify the difficulty of the habit. (If we have time)	-	Habit Difficulty Setting Test	M
13	AUTH-USER	A user shall be the only one that can access their personal profile.	-	Auth-User Test	M
14	UNIQHAB	Users shall be able to create unique habits by describing a habit and time to do the habit.	-	Unique Habit Test	M
15	IDTAG	A user shall have a unique id tag.	-	ID tag Test	M
Non-Functional					
	Identifier	Requirements	Use Case	Test Case	Type
1	DATABASE	The applications database is firebase.	-	Database Test	M
2	C.RESPONSE	Client side login should take less than 10 seconds under optimal circumstances.	-	Client Login Response Time Test	M
3	S.RESPONSE	Server Side login should take less than 5 seconds under optimal circumstances.	-	Server Login Response Time Test	M
4	FLUX	The application should be based on the FLUX pattern.	-	MVC Pattern Test	M
5	MULTCATEG	The application should have more than two categories for habits.	-	Multiple Categories Test	M
6	ADDCATEG	The application should allow a habit to be added to only one category.	-	Add Habit to Category Test	M
7	EMAILREM	The application should send an email at most 5 minutes after failing to do a habit.	-	Email Reminder Test	M



Functional

Functional Requirements are the functional aspects of the software.

SIGNUP - Sign up new User

Sign up new user Test	
Requirement	SIGNUP
Use Case	UC1 - Get Started
Scenario	The form should accept the request for sign up and redirect the user to login / personal profile if automatic login is enabled.
Precondition	The website is open, the no user is currently signed in, and the user does not already have an account.
Test Steps	<ol style="list-style-type: none">1. Click on 'Sign up'.2. Enter personal details:<ol style="list-style-type: none">a. First name enter 'John'.b. Last name enter 'Doe'.c. Age enter '36'.d. Sex enter 'Male'.e. Email enter 'johndoe@gmail.com'.f. Password enter '123veryGood'.g. Confirm Password enter '123veryGood'.3. Click 'Submit'.
Expected Results	The user is redirected to their Personal Profile with the correct personal details being displayed.
Comments	-



LOGIN - Login Existing User

Login existing user Test	
Requirement	LOGIN
Use Case	UC2 - Login
Scenario	The form should accept the request for login and redirect the user to their personal profile.
Precondition	The website is open, the user is currently signed in, and the user already has an account.
Test Steps	<ol style="list-style-type: none">1. Click on 'Login with existing account'.2. Enter personal details:<ol style="list-style-type: none">a. Email enter 'johndoe@gmail.com'.b. Password enter '123veryGood'.3. Click 'Login'.
Expected Results	Users are redirected to their Personal Profile with the correct personal details being displayed.
Comments	-

LOGOUT - Logout Existing User

Logout existing user Test	
Requirement	LOGOUT
Use Case	UC3 - Logout
Scenario	The site should accept the request for logout and redirect the user to the home page.
Precondition	The website is open, the user is currently signed in.
Test Steps	<ol style="list-style-type: none">1. Click on *Profile Picture* in the top-right corner.2. Click 'Logout' or 'Sign out'.
Expected Results	User is successfully logged out & is redirected to the home page of the website.
Comments	-



CREATEHAB - Create Habit

Create Habit Test	
Requirement	CREATEHAB
Use Case	UC4 - Create a new Habit
Scenario	The user can create personal habits with a name describing the habit and with additional custom information.
Precondition	The website is open, the user is currently signed in and can view the 'My Habits' header of the profile page.
Test Steps	<ol style="list-style-type: none">1. Click on the header 'My Habits'.2. Click on 'Create new Habit'.3. Fill in information about the new habit:<ol style="list-style-type: none">a. Habit name enter 'Morning Jog'.b. Create a new Category (See CATHAB Test).c. Add to Category (See ADDHAB Test).d. Frequency enter 'Every day'.e. Set Reminder enter '6:30 am'.4. Click 'Finalize'.
Expected Results	User has successfully created a habit and is redirected to 'My Habits' page where they can view their current and newly created habit(s).
Comments	Subject to change*



CATEGHAB- Categorize Habit

Create Category Test	
Requirement	CATEGHAB
Use Case	UC4 - Create a new Habit
Scenario	The user can create personal categories with a label describing the containing habits.
Precondition	The website is open, the user is currently signed in and can view the 'My Habits' header of the profile page.
Test Steps	<ol style="list-style-type: none">1. Click on the header 'My Habits'.2. Click on 'Create new Category'.3. Fill in information about the new category:<ol style="list-style-type: none">a. Category label enter 'Exercise'.b. Color set to 'Red'.4. Click 'Finalize'.
Expected Results	User has successfully created a category and is redirected to 'My Habits' page where they can view their current and newly created habit(s) in their respective category.
Comments	Subject to change*



ADDHAB - Add Habit to Category

Add Habit to Category Test	
Requirement	ADDHAB
Use Case	UC4 - Create a new Habit
Scenario	The user should be able to add a habit to a pre-existing or newly created category.
Precondition	The website is open, the user is currently signed in and can view the 'My Habits' header of the profile page.
Test Steps	1. (See CREATHAB Test steps 1 - 3b). c. Set Category select 'Exercise'. d. (See CREATHAB Test steps 3d - 4). 4. Click 'Finalize'.
Expected Results	The user has successfully added their habit to a category and after finalization is redirected to 'My Habits' page where they can view their current and newly created habit(s) in their respective category.
Comments	Subject to change*



REMINDER - Send Email Reminder

Send Email Reminder Test	
Requirement	REMIND
Use Case	-
Scenario	The user should receive an Email Notification/Reminder whenever they have missed a scheduled habit. I.e. 'Eat breakfast at 7:00am' the user will receive a reminder at the latest at 7:05am that they should eat breakfast now.
Precondition	The user has a scheduled habit, at 'Monday 7:00am', named 'Healthy Breakfast' which they have not marked completed in 'My Habits'.
Test Steps	<ol style="list-style-type: none">1. Receive an Email Notification at '7:05am' on Monday.2. Click the link in the Email, redirecting you to the website.3. Log in (See LOGIN Test).4. Once redirected to Personal Profile:<ol style="list-style-type: none">a. Navigate to 'My Habits'.b. Select the habit 'Healthy Breakfast'.c. Click the checkbox on the left hand side of the habit and mark it as completed.
Expected Results	User has successfully completed the habit 'Healthy Breakfast' at approximately '7:07am' on Monday.
Comments	Subject to change*



PROFILE - User Profile

User Profile Test	
Requirement	PROFILE
Use Case	-
Scenario	Whenever the user is logged in they should be able to view their Personal Profile.
Precondition	The user created an account and is logged in on the website, viewing the Home Page.
Test Steps	<ol style="list-style-type: none">1. Click your user profile picture in the TOP-RIGHT corner.2. Click on 'My Profile'.3. Redirect to 'myProfile'.
Expected Results	User have successfully been redirected to 'myProfile' and can view all of their personal information as well as navigate to 'My Habits'.
Comments	Subject to change*



EDITDETAILS - Edit Personal Details

Edit Personal Details Test	
Requirement	EDITPD
Use Case	-
Scenario	The user wishes to edit their personal information such as name, email etc.
Precondition	The user is logged in and currently in their Personal Profile. The current email information is invalid and needs to be updated.
Test Steps	<ol style="list-style-type: none">1. Click 'Personal Details'.2. Click 'Edit'.3. Navigate to the email field.4. Click on 'Email-address'.<ol style="list-style-type: none">a. Enter 'JohnDoe@gmail.com'.5. Navigate to the bottom of the page.<ol style="list-style-type: none">a. Click 'Finalize Changes'.b. In the popup window, click 'Confirm'.
Expected Results	User has successfully changed their email address information from 'johndoe@gmail.com' to 'JohnDoe@gmail.com'.
Comments	<p>There needs to be a secondary 'FUNCTIONAL' email address we can use in place of changing an uppercase letter.</p> <p>Subject to change*</p>



VIEWDETAILS - View Personal Details

View Personal Details Test	
Requirement	VIEWPD
Use Case	-
Scenario	The user wishes to view their personal information such as name, email etc.
Precondition	The user is logged in and currently in their Personal Profile.
Test Steps	1. Click 'Personal Details'.
Expected Results	User has successfully navigated to 'Personal Details' and can fully view all details (except password).
Comments	

PROGRESS - Current Progress

Current Progress Test	
Requirement	PROGRESS
Use Case	-
Scenario	The user wishes to view their personal progress.
Precondition	The user is logged in and has completed one or multiple habits.
Test Steps	1. Navigate to the TOP-RIGHT corner. 2. View under/right of Profile Picture. 3. Current Progress: a. Progress Bar = 1150 / 2500. b. Streak Multiplier = x1.15.
Expected Results	The user can successfully view their Progress after completing a Habit.
Comments	Values in Test Steps are preliminary and does not reflect the test scenario accurately. Not finalized if this Test Case is relevant. Subject to change*



DIFFICULTY - Habit Difficulty Setting

Habit Difficulty Setting Test	
Requirement	DIFFICULTY
Use Case	-
Scenario	The user wishes to specify how difficult this habit will be to complete for them personally.
Precondition	The user is logged in and wants to add a difficult habit.
Test Steps	1. (See CREATHAB Test steps 1 - 3e). 3. Current Progress: f. Set Difficulty enter 'Hard'. 4. Click 'Finalize'.
Expected Results	User can successfully add a difficulty of 'Hard' to their Habit.
Comments	Values in Test Steps are preliminary and does not reflect the test scenario accurately. Not finalized if this Test Case is relevant. Subject to change*



AUTH-USER - User Authentication

Auth-User Test	
Requirement	AUTH-USER
Use Case	-
Scenario	Another user tries to access your personal profile with your personal link.
Precondition	You have an account. Someone with another account has access to your personal link. You're logged in to the user without access to the personal profile, but with the link.
Test Steps	1. Enter the personal link: "iHabitMe.firebaseio.com/#User1092" 2. Press Enter. 3. Get redirected to: " iHabitMe.firebaseio.com/#Home ".
Expected Results	The user can not view the personal profile of another user. And is successfully redirected to the home page.
Comments	The precondition is long due to difficulty describing the conditions. Subject to change*



UNIQHAB - Unique Habit Creation

Unique Habit Test	
Requirement	UNIQHAB
Use Case	-
Scenario	A user wants to create an unique habit by describing a habit and a time to do the habit.
Precondition	The user has an account & is logged in.
Test Steps	1. (See CREATEHAB 1 - 4)
Expected Results	User has successfully created a habit and is redirected to 'My Habits' page where they can view their current and newly created habit(s).
Comments	Subject to change*

IDTAG - Unique User ID

ID tag Test	
Requirement	IDTAG
Use Case	-
Scenario	A users username should have an unique tag or unique id.
Precondition	The user has created an account (See SIGNUP). Access to database userdata.
Test Steps	1. Open Database application. 2. Navigate to userdata. 3. View users ID tag.
Expected Results	The user has an unique ID tag.
Comments	This test is abstract due to the fact we are using a third party database system. Subject to change*



Non-Functional

Non-functional requirements are the requirements which do not relate to the overall function of the software and are implicit or expected characteristics of the system which users can make assumptions about.

DATABASE - Database

Database Test	
Requirement	DATABASE
Use Case	-
Scenario	The web server is connected to Firebase and stores all relevant data about the user and updates in the cloud storage.
Precondition	The web server contains no breaking bugs and errors and the database structure is predefined and not preliminary to be changed in the near future.
Test Steps	-
Expected Results	-
Comments	Subject to change*



C.RESPONSE - Client Login Response Time

Client Login Response Time Test	
Requirement	CRESPONSE
Use Case	-
Scenario	When logging in with under optimal conditions, such as a stable internet connection, decent hardware, no heavy load on database and server, the login should take less than 10 seconds to be confirmed for the client.
Precondition	Internet Connection is stable at 10/10Mbps upload and download speed, connected through Ethernet Cable. Database is under no external load other than the tester currently sending a login request.
Test Steps	1. (See LOGIN Test).
Expected Results	Login is successful within the 10 second limit.
Comments	Subject to change*

S.RESPONSE - Server Login Response Time

Server Login Response Time Test	
Requirement	SRESPONSE
Use Case	-
Scenario	When logging in with under optimal conditions, such as a stable internet connection, decent hardware, no heavy load on database and server, the login should take less than 5 seconds to be registered and confirmed by the Server.
Precondition	Internet Connection is stable at 10/10Mbps upload and download speed, connected through Ethernet Cable. Database is under no external load other than the tester currently sending a login request.
Test Steps	1. (See LOGIN Test).
Expected Results	Login is successful within the 5 second limit.



Comments	Subject to change*
----------	---------------------------

MVCPATTERN - MVC Pattern

MVC Pattern Test	
Requirement	MVCPATTERN
Use Case	-
Scenario	When writing code the code should follow the Model, View, Controller Pattern.
Precondition	You are at the top of the hierarchy of the Projects folder structure. ("src").
Test Steps	<ol style="list-style-type: none">1. Go to package: 'webserver'.2. Go to package: 'view'.<ol style="list-style-type: none">a. Verify that 'view' only contains messages and no calculations in any classes.3. Return to package: 'webserver'.4. Go to package 'model'.<ol style="list-style-type: none">a. Verify that 'model' only contains interfaces and abstract classes.5. Return to package: 'webserver'.6. Go to package: 'controller'.<ol style="list-style-type: none">a. Verify that 'controller' the only package containing calculations and calls to the 'view' & 'model' packages?
Expected Results	The Project follows the MVC Pattern without fault and is properly structured.
Comments	Subject to change*



MULT.CATEG. - Multiple Categories

Multiple Categories Test	
Requirement	MULTCATEG
Use Case	-
Scenario	The user should be able to create and store multiple habits across multiple created and predefined categories.
Precondition	A user is logged in and on the 'My Profile' Page. There is already two predefined or user-created categories.
Test Steps	1. See (Create Category Test steps 1 - 4).
Expected Results	The (Third) new category should be functioning just as good as the previous two categories and can be viewed from the 'My Habits' Page.
Comments	Subject to change*



ADDCATEG. - Add Habit to Categories

Add Habit to Category Test	
Requirement	ADDCATEG
Use Case	-
Scenario	The user should be able to add a habit to a pre-existing or newly created category.
Precondition	The website is open, the user is currently signed in and can view the 'My Habits' header of the profile page.
Test Steps	1. (See CREATHAB Test steps 1 - 3b). c. Set Category select 'Exercise'. d. (See CREATHAB Test steps 3d - 4). 4. Click 'Finalize'.
Expected Results	User has successfully added their habit to a category and after finalization is redirected to 'My Habits' page where they can view their current and newly created habit(s) in their respective category.
Comments	Subject to change*



EMAILREM - Email Reminder

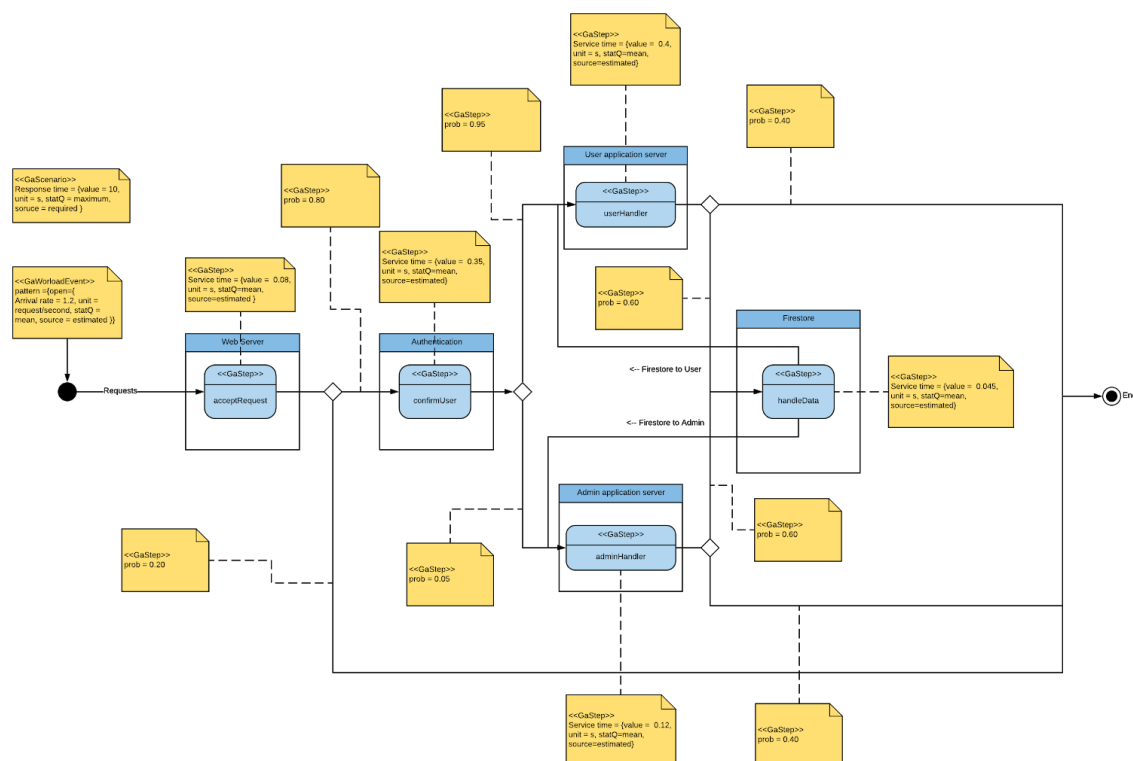
Email Reminder Test	
Requirement	EMAILREM
Use Case	-
Scenario	The user should receive an Email Notification/Reminder whenever they have missed a scheduled habit. I.e. 'Eat breakfast at 7:00am' the user will receive a reminder at the latest at 7:05am that they should eat breakfast now.
Precondition	The user has a scheduled habit, at 'Monday 7:00am', named 'Healthy Breakfast' which they have not marked completed in 'My Habits'.
Test Steps	<ol style="list-style-type: none">1. Receive an Email Notification at '7:05am' on Monday.2. Click the link in the Email, redirecting you to the website.3. Log in (See LOGIN Test).4. Once redirected to Personal Profile:<ol style="list-style-type: none">a. Navigate to 'My Habits'.b. Select the habit 'Healthy Breakfast'.c. Click the checkbox on the left hand side of the habit and mark it as completed.
Expected Results	User has successfully completed the habit 'Healthy Breakfast' at approximately '7:07am' on Monday.
Comments	Subject to change*

Performance Requirement Modelling with MARTE

In this section there will be an Activity diagram on the system that will be created, we will set estimated service time values on the activities and required time for the system to respond using MARTE. The service time values and probability percentage is needed for the D2 section Performance modelling so that is why we have to estimate their values in this section since MARTE and Performance modelling in D2 is related to each other.

Activity diagram

This is the estimated visual of the system and the estimated values. From the requirements we believe the system will look similarly to this activity diagram, we will enter the web server and then the user tries to either authenticate or leave the web page. We estimate that the user leaves around 20 % of the time. If the user choose to authenticate they will either log in as a user or an admin. We believe the probability for logging in as a user is 95 % and 5 % as an admin. The user then does what they want, create daily habits and categories, then save those new objects in the database Firestore. When the user feels done they exit the application. The admin works with firestore back and forth until they are done and then exit the application. We estimate that the user create approximately 3 things everytime they are logged in and the same for admin. This is how we believe the system will be run.





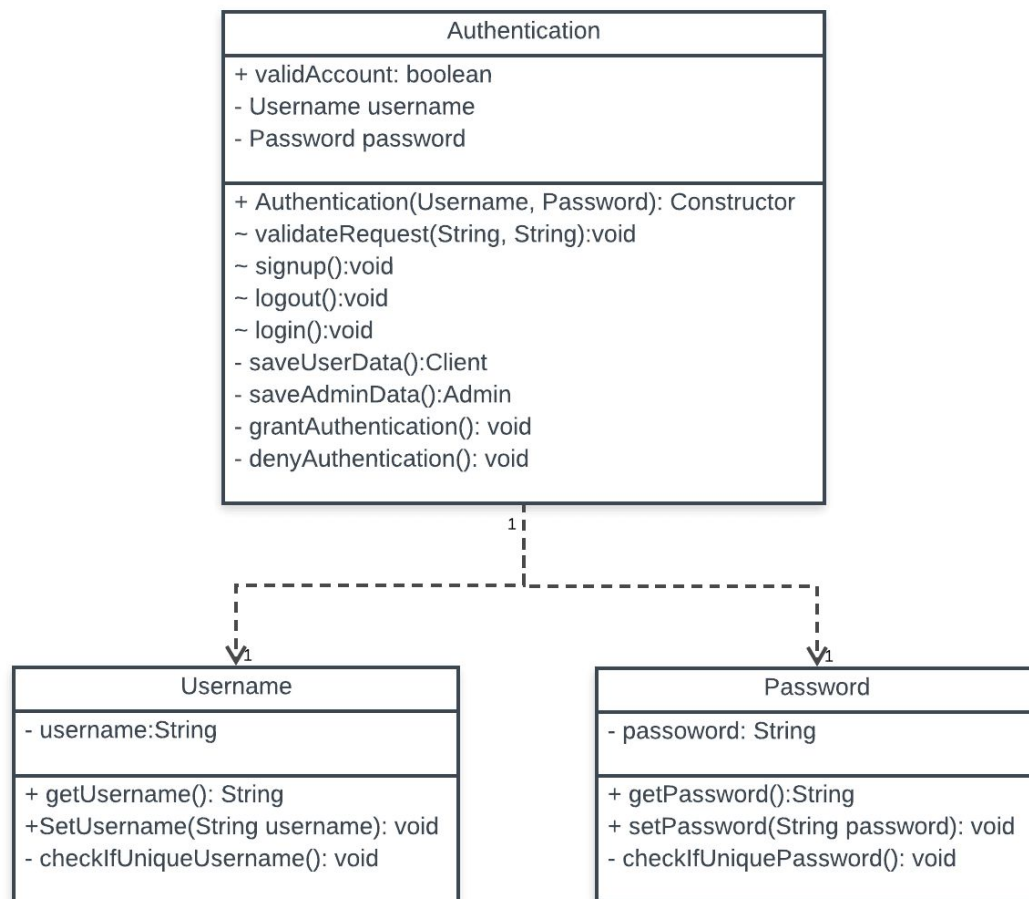
Model UML Requirements

Class Diagrams

This section will visualize the requirements in the application with class diagrams.

LOGIN

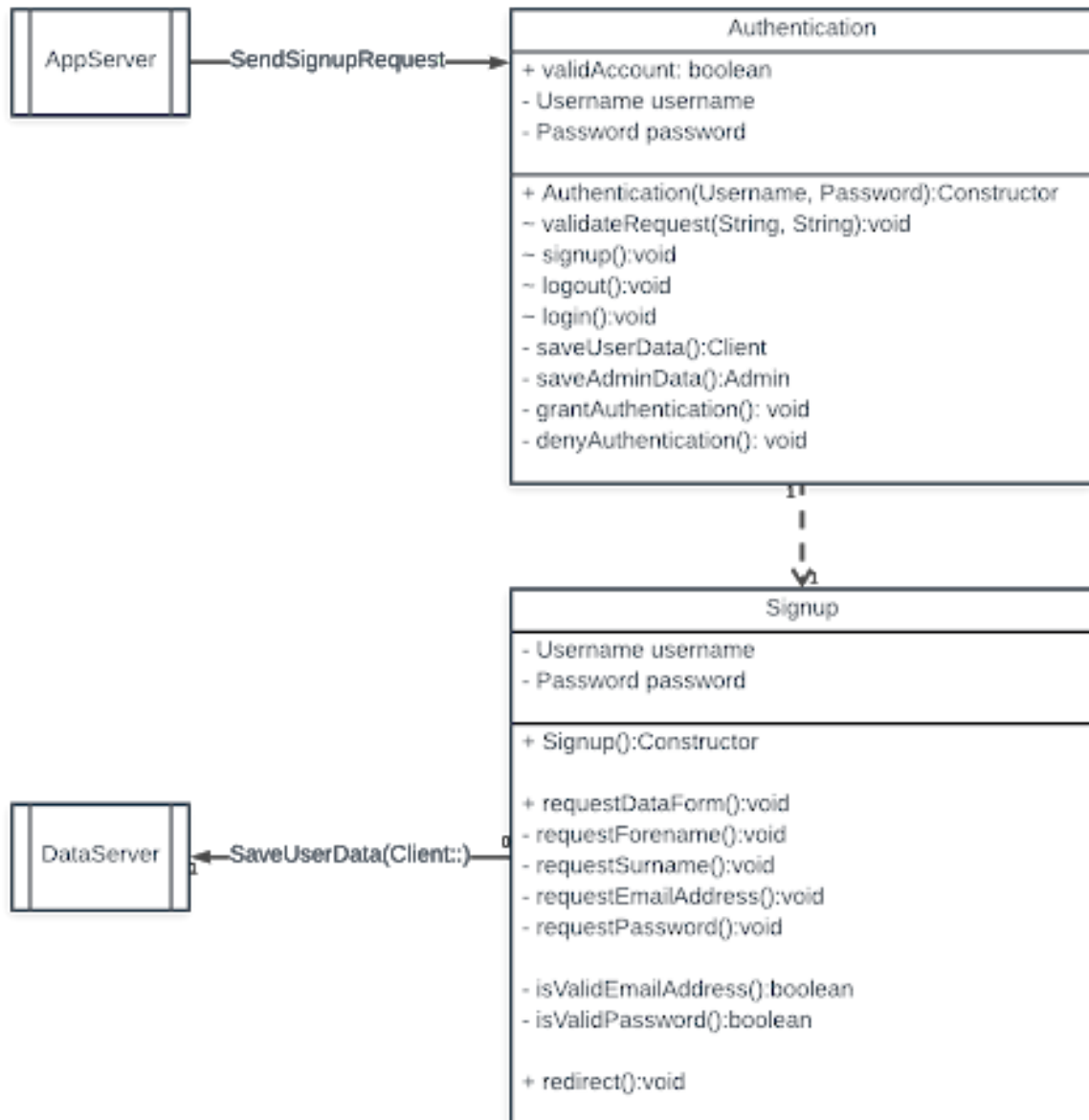
Below is the class diagram for the requirement LOGIN - Requirement.





SIGNUP

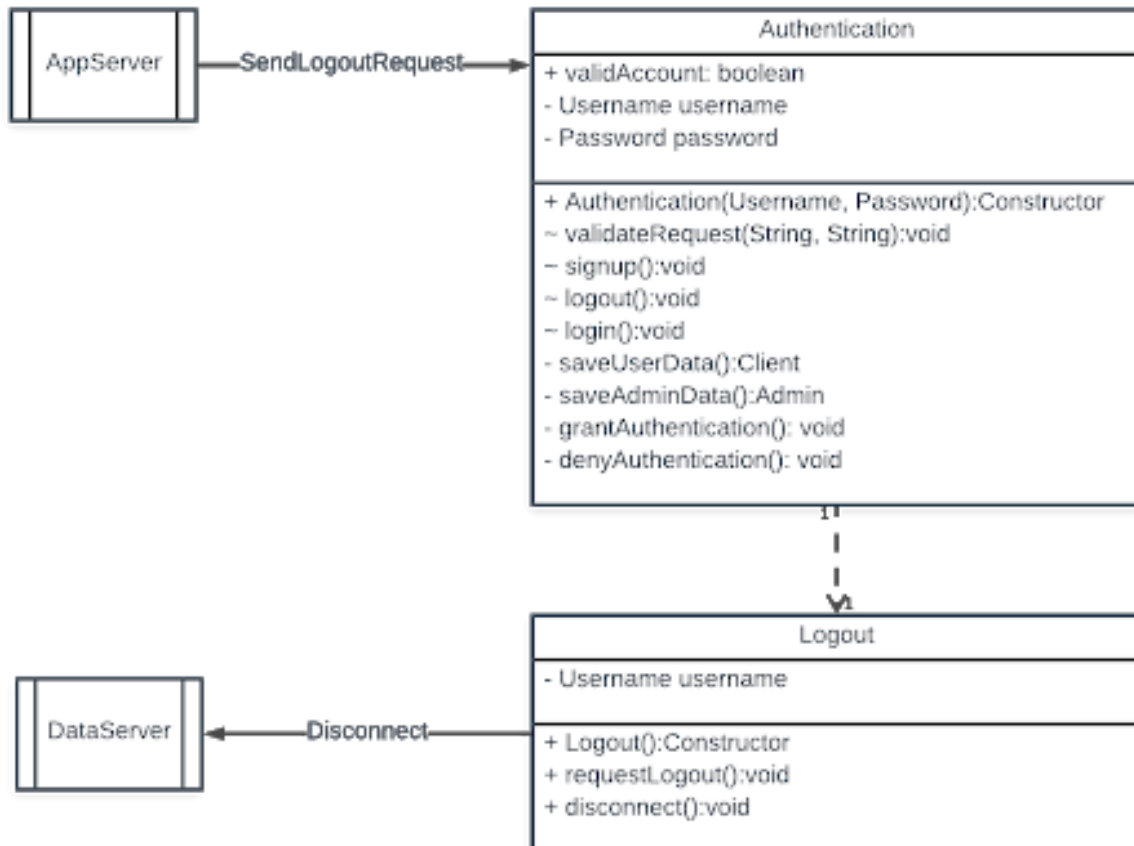
This is the UML Class Diagram for the SIGNUP - Requirement.





LOGOUT

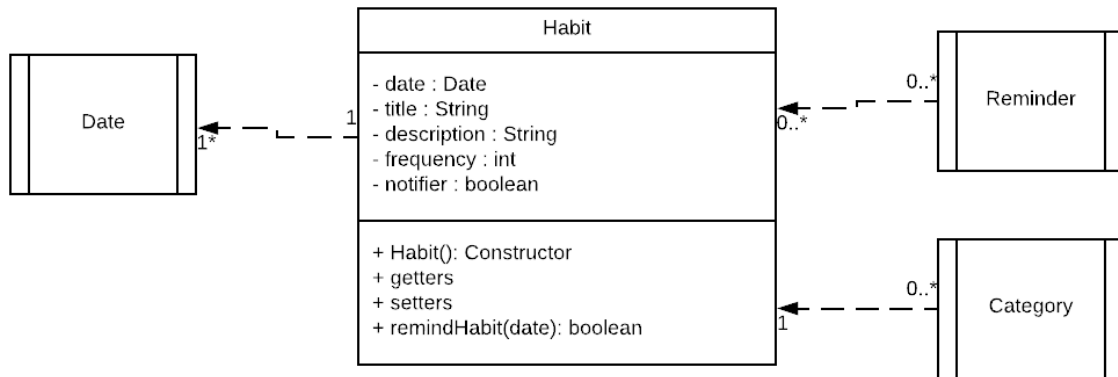
This is the UML Class Diagram for the LOGOUT - Requirement.





Habit

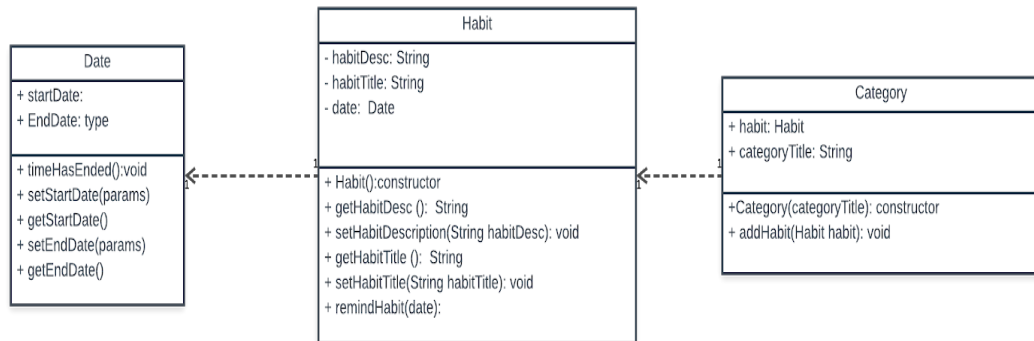
The requirements HABIT, UNIQHABIT, DATE, TITLE, DESCRIPTION, FREQUENCY and NOTIFIER represented as a Class diagram.





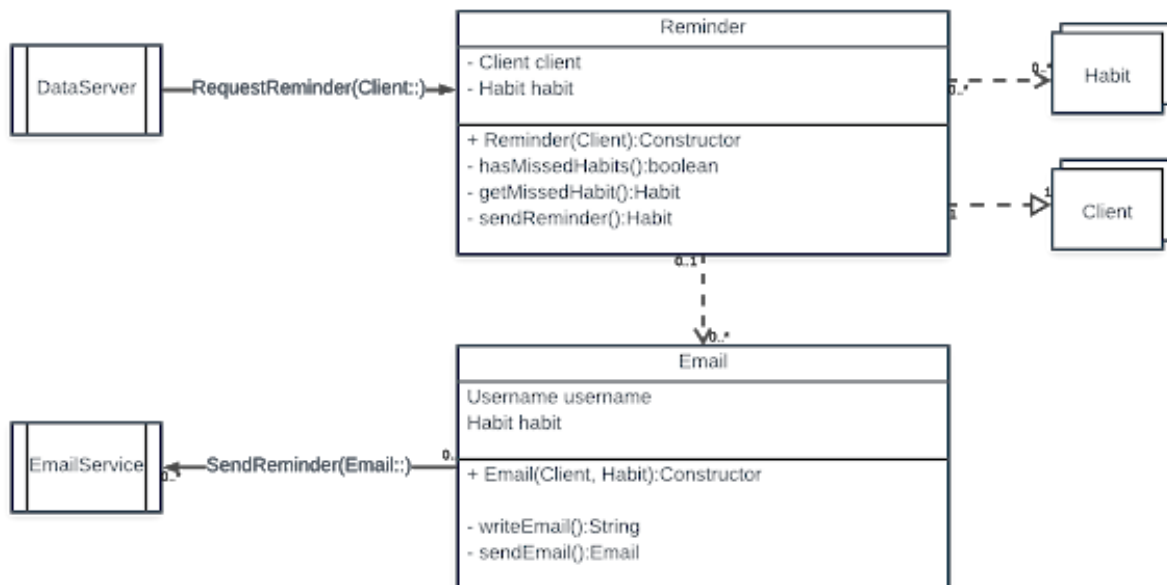
ADDHAB

Below is the class diagram for the requirement ADDHAB - Requirement.



REMIND

This is the UML Class Diagram for the REMIND - Requirement.





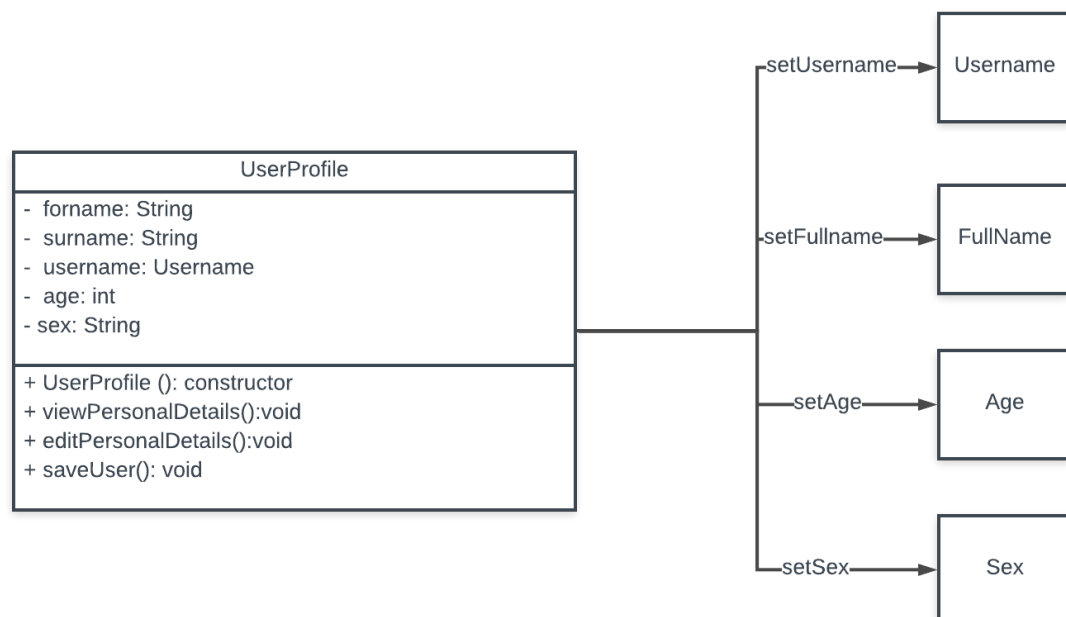
CREATHAB

This is the UML Class Diagram for the CREATHAB - Requirement.



EDITPD

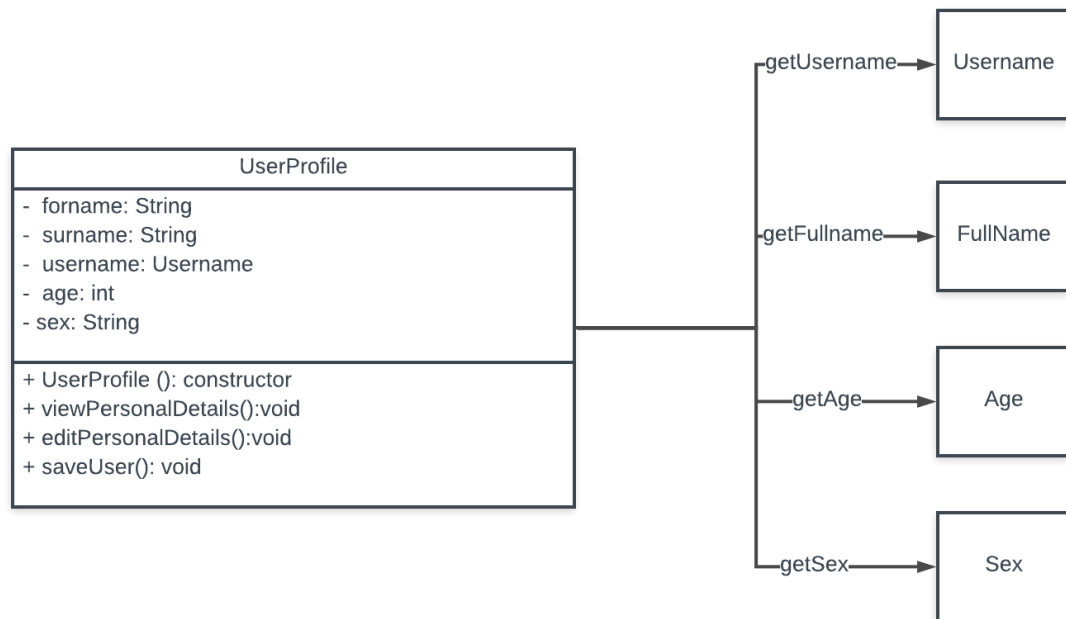
This is the UML Class Diagram for the EDITPD - Requirement.





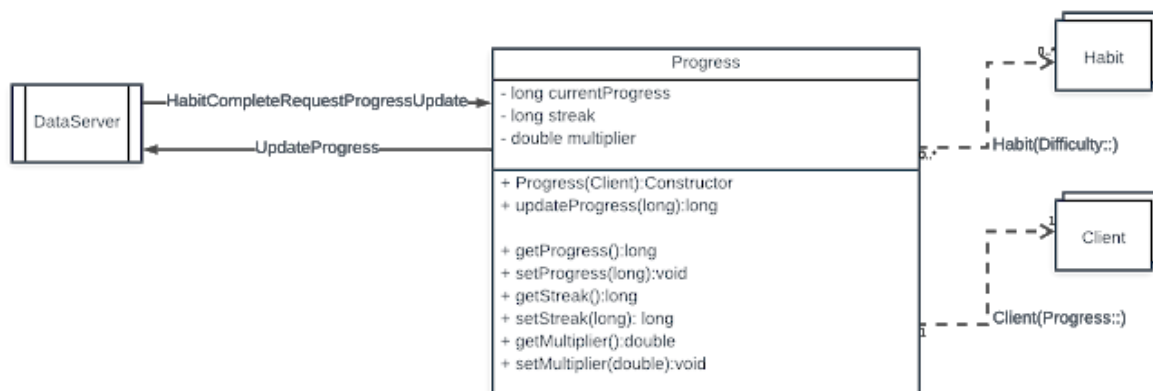
VIEWPD

This is the UML Class Diagram for the VIEWPD - Requirement.



PROGRESS

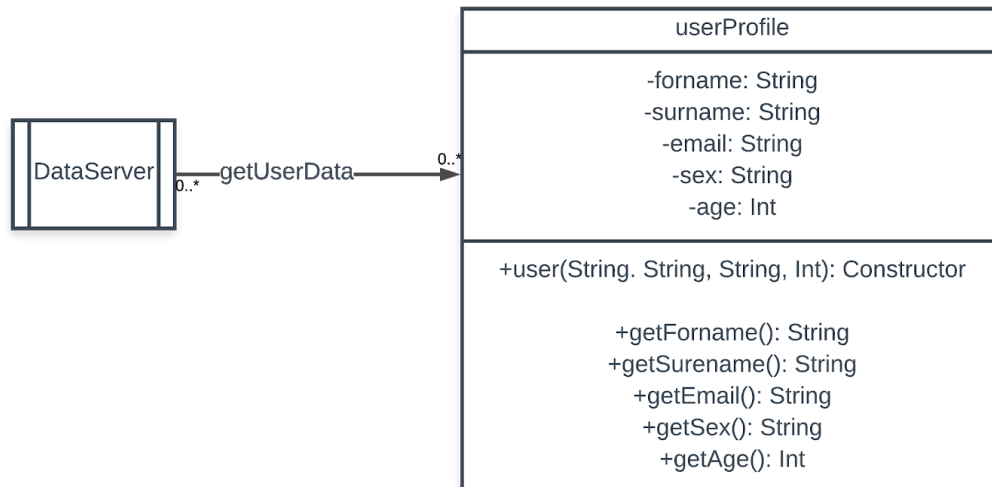
This is the UML Class Diagram for the PROGRESS - Requirement.





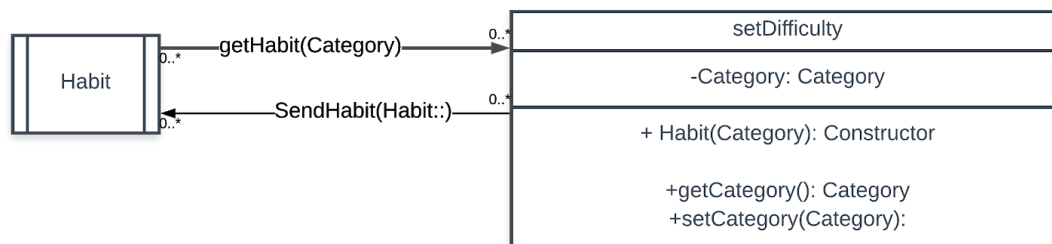
USERPRO

Below is the class diagram for the USERPRO requirement.



CATEGHAB

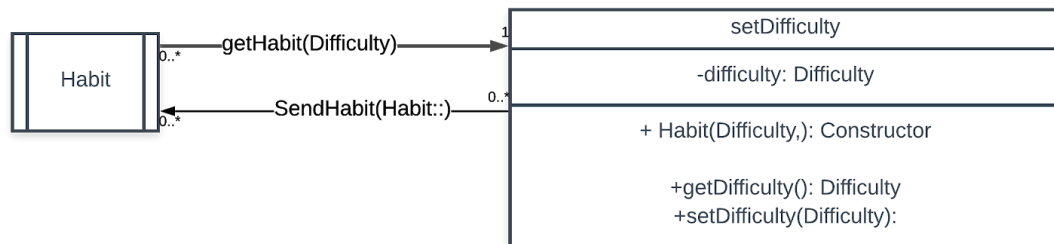
Below is the class diagram for the CATEGHAB requirement.





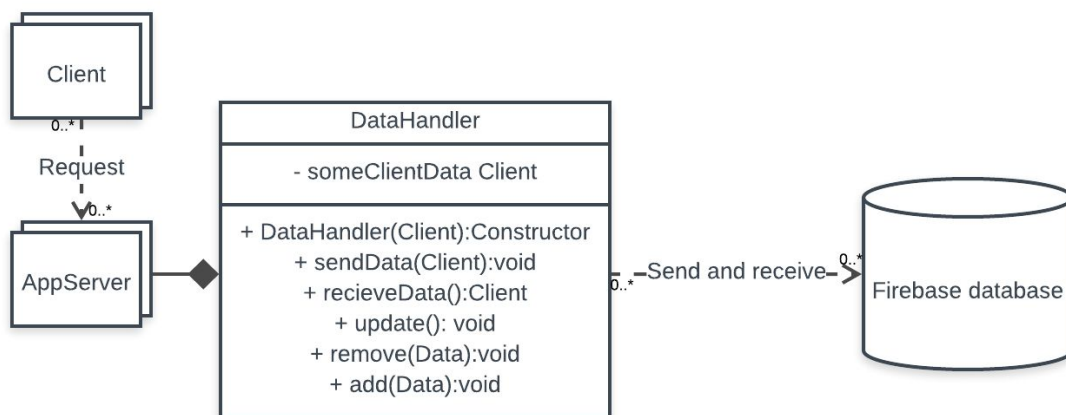
HABDIFF

Below is the class diagram for the HABDIFF requirement.



FIREBASE

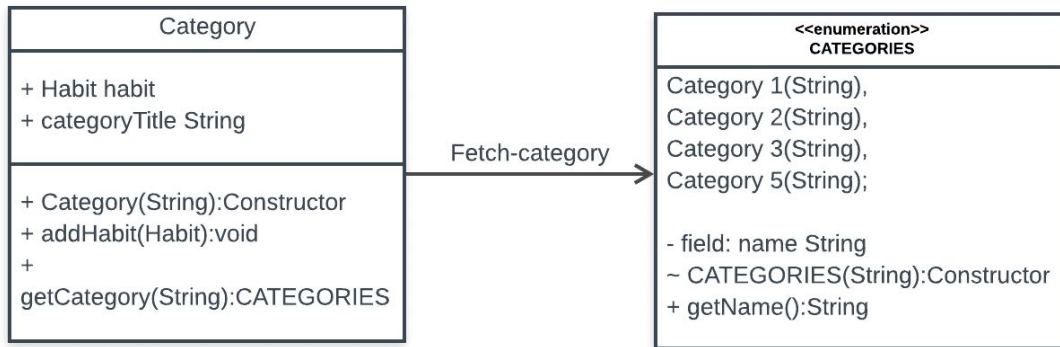
Below is the class diagram for the FIREBASE requirement.





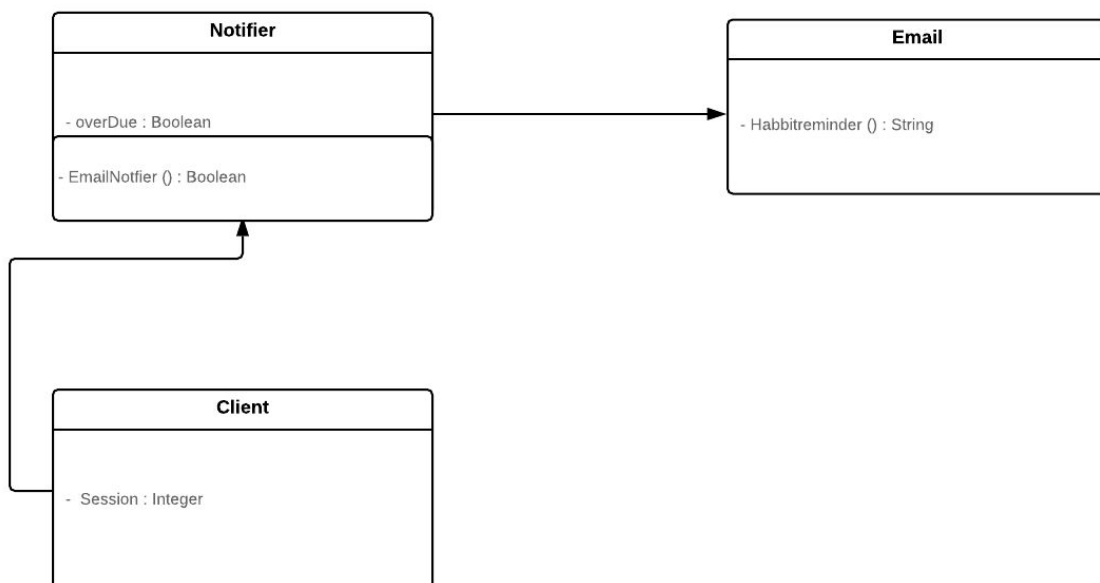
MULT.CATEG

Below is the class diagram MULT.CATEG requirement.



EMAILREM

Below is the class diagram of EMAILREM requirements.

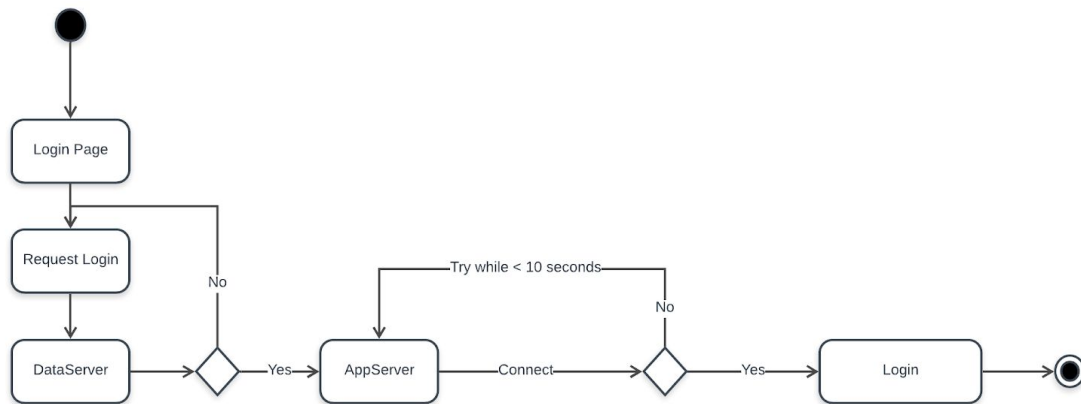




State/Activity Diagrams

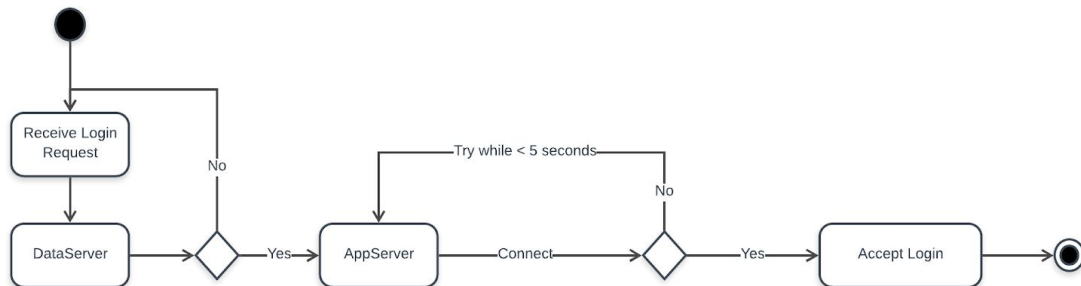
C.RESPONSE

Below is the state activity diagram for the C.RESPONSE requirement.



S.RESPONSE

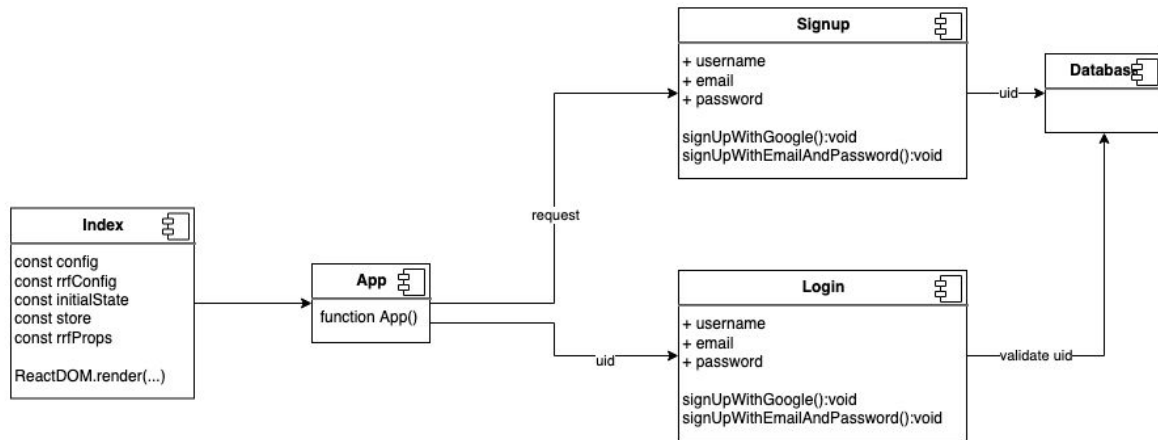
Below is the state activity diagram for the S.RESPONSE requirement.





ID-TAG

Below is the state activity diagram for the ID-TAG requirement.



AUTH-USER

Below is the state activity diagram for the AUTH-USER requirement.

