

CSE 343

SOFTWARE ENGINEERING COURSE

FINAL PROTOTYPE REPORT



PROJECT NAME: **NEED++**

TEAM NAME: **TEAM ARI**

TEAM MEMBERS:

ESRA NUR ARICAN (Scrum Master)

ELİF KELES (Product Owner)

SEMA DİLBER

MERVE DUR

MEDİNE ASLAN

Index

1. DEFINITION OF THE PROJECT	3
2. USER STORY AND ITS RELATED SCENARIOS	3
2.1 First User Story.....	3
2.2 Second User Story.....	5
3. PROJECT BACKLOG DOCUMENT	6
3.1 PRODUCT BACKLOG with TASKS and SUBTASKS	6
3.2 PRODUCT BACKLOG WITH USER STORIES.....	10
3.3 PRODUCT BACKLOG with BUGS	11
4. UML DIAGRAMS.....	12
4.1 CONTEXT MODEL	12
4.2 PROCESS MODEL.....	13
4.3 USE CASE DIAGRAMS and Its RELATED TABLES	14
4.4 SEQUENCE DIAGRAMS.....	18
4.5 CLASS DIAGRAM	21
4.6 STATE DIAGRAMS	22
5. SYSTEM ARCHITECTURE DOCUMENT	22
6. WORKING GRAPHICAL INTERFACE AND BACKEND SERVICES	25
7. TEST REPORTS	29
8. NEED++ PROJECT REQUIREMENT ANALYSIS	37
8.1. PREFACE	37
8.2. INTRODUCTION and SCOPE OF THE PROJECT	37
8.3. FUNCTIONAL REQUIREMENTS.....	38
8.3.1 USER REQUIREMENTS.....	38
8.3.2 SYSTEM REQUIREMENTS	39
8.4. NON-FUNCTIONAL REQUIREMENTS	39
9. CONTINUOUS INTEGRATION and CONTINUOUS DELIVERY CONCEPTS	41
10. ACCESSIBILITY OF THE PROJECT TO THE POTENTIAL USERS	44
11. REFERENCES	46
12. SOME LINKS ABOUT NEED++ PROJECT	46

1. DEFINITION OF THE PROJECT

The web application called Need ++, which we plan to design within the scope of our software engineering course, is an online platform that we plan to bring together the needy and benefactors without anyone in between. With this web application, individual or collective assistance requests can be made and people who want to meet them can deliver their help to those in need.

Basically, the main features planned to be included in our application are as follows:

- Declaration of need
- Need fulfillment

With the requirement announcement feature, you can share anything you need on this platform on any subject. For example, baby clothes, stationery, etc.

You can also support those in need with the need-solving feature, in matters that you think you can help from among the postings shared on the site.

In order to use these features, both sides of the system must first be registered. Of course, we do not share the information of our users with anyone in order to ensure the security of the information regarding the confidentiality of the help calls.

To mention some of the features that we think to add as we develop our project; We plan to enrich our application with features such as categorizing shared need notices, adding search feature by keywords, adding urgency classification and perhaps adding a map view in the future.

With this application idea, which we consider to be an important project in the name of social responsibility and volunteering, our main goal is to provide assistance to many people.

2. USER STORY AND ITS RELATED SCENARIOS

2.1 First User Story

Posting a need to Need++

Kevin is an animal lover who feeds and takes care of the cats and dogs living on the wasteland in his neighborhood. Realizing that the animals start to get cold with the arrival of

winter, Kevin thinks that animals need huts and decides to publish a post in Need ++ app, which he had heard of before, because he thought he could not buy it alone.

To do this, Kevin open the Need ++ system in his computer and tries the publish a need post. However, he is directed to the page he sees that he needs to log in and it understands that it is necessary to register in the system in order to add a need. Therefore, he registers to the system and after that he sees the homepage and posts.

Kevin repeats what he wants to do after the sign in; adding a need post. He fills the necessary information about the need a very clear and descriptive way for people to understand easily and adds this need posting to the system. So, the system saves this posting that Kevin entered in the database that is among other postings and he hopes someone can see and get that need quickly.

Related scenario

Initial assumption: A user has one or more need she/he wants to announce that to other people. She/he has successfully logged on to Need++.

Normal: The user chooses the sign up and is prompted to enter name, surname, birthday, e-mail, username, password, again password, country, city and district to signs up to the system. On completion of the sign up, the system automatically directs the user to homepage as logged in. After that the user chooses “add need” for posting her/his need in the system and is prompted to enter tittle of need, location and well and clearly defined need and posting completes. After this process the system automatically directs the user to homepage too.

What can go wrong: The user enters an invalid password at the sign-up stage. She/he should be saw a warning message about that.

The user enters two passwords that are not the same enter at the sign-up stage. She/he should be saw a warning message about that.

The user enters an invalid date enter at the sign-up stage. She/he should be saw a warning message about that.

The user enters an invalid e-mail address enter at the sign-up stage. She/he be saw a warning message about that

The user enters an invalid location (invalid country or/and city or/and district) enter at the sign-up stage. She/he be saw a warning message about that.

Other activities: The user may see all need posts and number of active needs.

System state on completion: User is signed-up. The need post has been shared. The need post is visible to the other people.

2.2 Second User Story

Fulfill a need in Need++

Karen is an art teacher working at a school outside of Istanbul, Karen had previously posted a need posting in Need ++ as most students at her school didn't have a nice painting set, and a helpful businessperson responded and supported them. Karen was delighted with the incident, and she wanted to fulfill someone's need.

She logs in to Need++ and display the need post. While looking at the posts, the baby stroller post catches Karen's attention. When Karen reads the posting completely, she thinks that she can send the stroller, which she has not used since her daughter is growing, to this needy owner and sends an e-mail to the advertiser, Kate Malone, to the e-mail address `katemalone@gmail.com`, which he reached from her contact information. While logging out of the system, she realizes that the profile photo does not like it anymore and wants to change it. She chooses a new photo for her and logs out of the system.

Related scenario

Initial assumption: A user wants to fulfill other people's needs, he/she enter necessary login information in login page of the system. Then logging into Need++ system successfully.

Normal: The user looked need posts and he/she chose a need post. It displays the contact information that can be used to communicate with the person in need. Finally, he/she edited his/her profile photo and logged out.

What can go wrong: The user may have entered the wrong e-mail and password information while logging into the system the system shows notifications to the user to check his/her email address and password.

The user forgot his/her password, a current e-mail address of the user is requested, and a password is sent to this e-mail address. The user encounters the password reset screen where he/she creates a new password in accordance with the desired conditions and logs into the system.

The photo extension to be added while editing the profile photo may not be the desired extensions, the message that this extension is out of format is shown and asked to select the correct extension.

Other activities: The user may see all need posts and number of active needs.

System state on completion: User is logged on. The need posts have been displayed and choose one of them. Edit her/his profile photo then logged out.

NOTE: There is also a backlog table of user stories in the backlog part.

3. PROJECT BACKLOG DOCUMENT

Product backlog is a list of “to do” items that the Scrum team must tackle. They may be feature definitions for the software, software requirements, user stories, or descriptions of supplementary tasks that are needed, such as architecture definition or user documentation. (Sommerville/ Software Eng. /10th Ed.)

Due to this description, also with the help of Jira project management tool, we as Team Ari, created our product backlog with the tasks we want to tackle each week (or sprint) and some user stories.

Task Issues contain our weekly goals for the project progress. As we decide the goals, we like to break them into a few subtasks so that we can achieve them, as a team, easily. Also this makes everything easier to track and to be on time.

User Stories are basically us stepping into our user's shoes and think what else would we like to see in our project.

Bugs are about the problems we face when creating the project back-end or front-end. After we find a solution to get rid of the bug, we assign it as done.

3.1 PRODUCT BACKLOG with TASKS and SUBTASKS

ID	TASK	PRIORIT Y	STATUS	SPRIN T	NOTES	ASSINGNEE
1	Unit test research	High	Done	4		
2	Unit test creation	High	Done	4	JUnit is chosen as framework.	
3	Name- Surname- Password creation in the log-in part	Medium	Done	4		Esra
4	Checking the birthday and image file extension	Medium	Done	4		Medine
5	Changing the password	Medium	Done	4		Elif
6	Checking the city and country information	Medium	Done	4		Sema
7	Checking the validity of e-mail and phone number information	Medium	Done	4		Merve
8	Searching and learning about wireframe	Medium	Done	4		
9	Creating wireframe	Medium	Done	5		
10	Learn about COCOMO Calculation	High	Done	5		
11	Calculate COCOMO	High	Done	5		
12	Create and submit COCOMO Report	High	Done	5		Elif
13	Make a research about wireframes that similar to our project	Medium	Done	5		Esra

1 4	Search for a proper application to create a wireframe	Medium	Done	5		Esra
1 5	Draw the new pages of wireframe	Medium	Done	5		Sema
1 6	Starting of the log-in page	High	Done	5		Merve
1 7	Use Case Diagram Creation	Medium	Done	5		Medine
1 8	Choosing Framework & tools for the project development	High	Done	6	NetBeans IDE is chosen.	Medine
1 9	Research about html- java codes for project	High	Done	6		Sema
2 0	Creating to-do list for the recurring tasks	Medium	Done	6		Elif
2 1	Use case diagram(s) for the defined scenarios	Medium	Done	6		Medine
2 2	Creation of Presentation of First Prototype	High	Done	6		
2 3	UI creation (Starting the log-in page)	Medium	Done	6		Merve
2 4	Unit testing	High	Done			
2 5	Building	High	Done			
2 6	Testing	High	Done			
2 7	Deploy	High	In Progress			
2 8	Creating Personas	Low	Done		Not essential	
2 9	Creating product roadmap	Low	Done	7		Elif
3 0	Think about new testing fragments	Medium	Done	7		Elif
3 1	Create new branches on github repository	Medium	Done	8		Esra
3 2	Create a new Need++ project	Medium	Done	8		Medine
3 3	Research github desktop usage	Medium	Done	8		Merve
3 4	Research git usage	Medium	Done	8		Esra
3 5	Automatic test creation tool research -for frontend	Low	Done	8		Elif
3 6	UI- Continue	High	Done	8		Medine
3 7	Front-end test and test phases research	Low	Done	8		Merve
3 8	Unit test for front end	Medium	Done	9		Esra

3						
9	Search and learn about database	High	Done	9		Elif
4						
0	UI -Last details	High	Done	9		Sema
4						
1	Create tables for database	Medium	Done	9		Esra
4						
2	Connect database tables	Medium	Done	9		Esra
4						
3	Connect database with UI	Medium	Done	10		Merve
4						
4	Unit test -for bean class	Medium	Done	10		Esra
4						
5	Reunite database classes as one	Medium	Done	10		Merve
4						
6	Learn about creating a general database	Medium	Done	11		
4						
7	Sequence diagram creation	High	Done	11		Medine
4						
8	Class diagram creation	High	Done	12		Esra
4						
9	Project backlog document	High	Done	12		Elif
5						
0	System architecture document	High	Done	12		Elif
5						
1	Test Report	High	Done	12	Selenium and Junit tests	Merve and Elif
5						
2	Context Model	High	Done	12		Sema
5						
3	Writing error messages for sign-up page	Medium	Done	12		Esra
5						
4	Writing well defined user story and related scenerios	High	Done	12		Medine and Sema
5						
5	Process model	High	Done	12		Sema
5						
6	Use case diagrams and their detailed tables	High	Done	12		Medine
5						
7	State diagram creation	High	Done	12		Elif
5						
8	Creation of the Second Presentation Report	High	Done	12		
5						
9	Check if the password / data etc errors work correctly	Low	To Be Started			
6						
0	Add comments to code	Low	Done	13		Elif
6						
1	Unit test for bean and info class	Medium	Done	13		Elif
6						
2	Add a new feature for editing the need post	Medium	In Progress	15		
6						
3	Add a new feature for deleting the need post	Low	To Be Started			

6 4	Research and learn about Selenium	Medium	Done	13		Merve and Elif
6 5	Connect database tables with username	High	Done	13		Merve
6 6	Writing help page content	Medium	Done	13		Sema
6 7	Unit testing - continue	Medium	Done	14		Elif
6 8	Search for web services	Medium	Done	14		Sema
6 9	Updates on frontend	Medium	Done	14		Medine
7 0	Research for continuous integration and delivery	Medium	Done	15		Esra
7 1	Creating a document ab. cont. int. and delivery	Medium	Done	15		Esra
7 2	Research about web server	Medium	Done	15		Sema
7 3	Sequence diagram - update	Medium	Done	15		Medine
7 4	Unit tests - update	Medium	Done	15		Elif
7 5	Backlog document - update	Medium	Done	15		Elif
7 6	Accessibility documentation	Medium	Done	15		Sema
7 7	Getting a web host and domain	Medium	Done	15		Sema
7 8	Transfer project files with filezilla	Medium	Done	15		Sema
7 9	Forgot password page	Medium	In Progress			Sema
8 0	Edit profile page	Medium	Done	15		Elif
8 1	Show password as ****	Medium	Done	15		Merve
8 2	Process model - update	Medium	Done	15		Sema
8 3	Creating the 3rd report	Medium	Done	15		Esra
8 4	Use case diagram - update	Medium	Done	15		Medine
8 5	Use case table - update	Medium	Done	15		Medine
8 6	# of needs and needy info. incrementing on homepage display	Medium	Done	15		Merve
8 7	Class diagram - update	Medium	Done	15		Elif

3.2 PRODUCT BACKLOG WITH USER STORIES

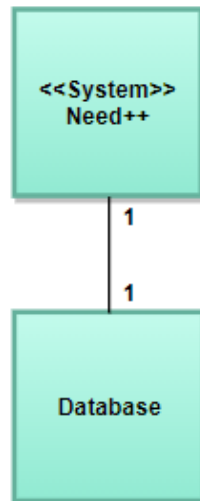
ID	AS A/AN	I WANT TO...	SO THAT	PRIORITY	STATUS	SPRINT	NOTES
1	User	create an account to Need++	I can use the app	Required	Done		
2	User	share my location for my need announcements	others can reach me	Required	Done	3	
3	User	a have a profile photo		Required	Done	3	
4	User	enter my e-mail info	I can start sharing my needs lists	Required	Done	3	
5	User	create and change my password		Required	Done	3	
6	User	sign up with my name and surname		Required	Done	3	
7	User	add my cell phone number for my need announcements	others can reach me	Required	Done	3	
8	User	have the ability to see my profile page and personal info	I can change my info if needed	Required	Done	10	
9	User	display other user's needs announcements	I can help others	Required	Done	11	
10	User	change my personal info	I can keep my personal info updated	Required	Done	13	
11	User	change my password if I forget it	I can log in to my Need++ account	Required	In Progress	14	
12	User	add my need list	other people can see my needs	Required	Done	11	
13	User	add support to other peoples needs		Required	To Be Started		
14	User	delete my need post		Non-urgent	To Be Started		
15	User	change/edit my need post		Non-urgent	In Progress	15	
16	User	be notified if my information is not valid	I can change to the correct info	Required	To Be Started		

3.3 PRODUCT BACKLOG with BUGS

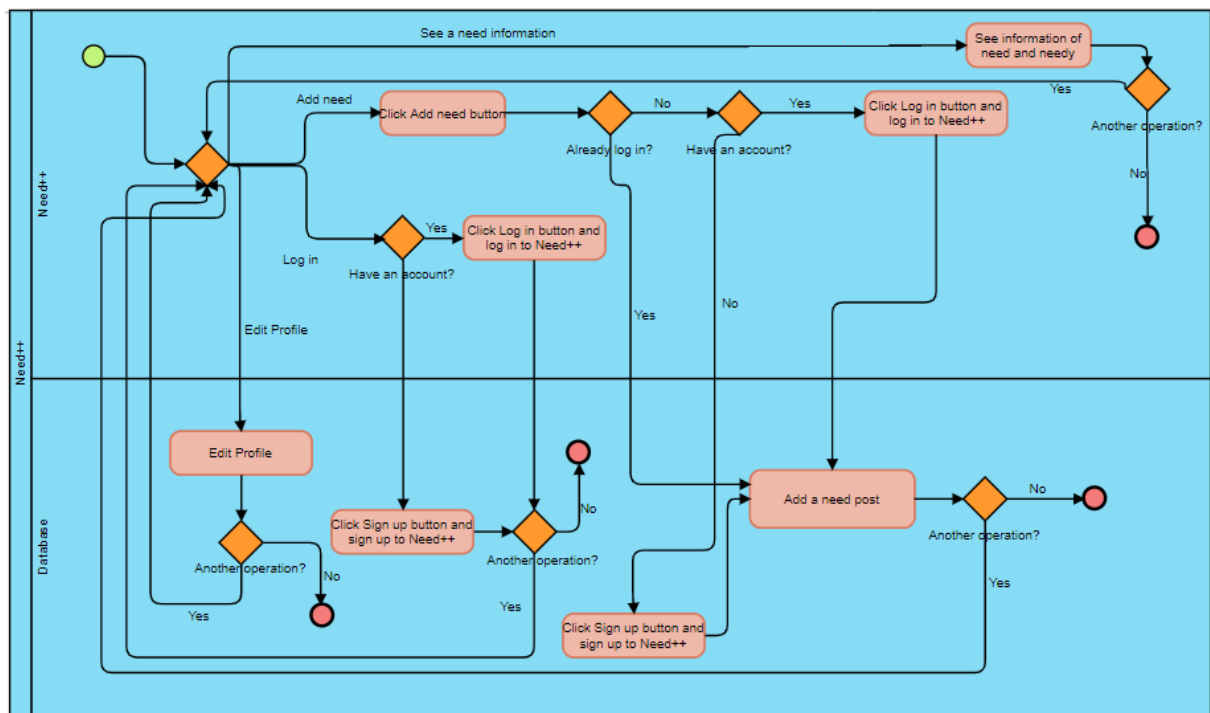
ID	BUG	PRIORITY	STATUS	SPRINT	NOTES	ASSINGNEE
1	Database- GUI connection	Medium	Done	9		Merve
2	Push- Pull local changes on database	Medium	To Be Started	9		
3	Working with database from different computers	Medium	Done	9		Esra
4	Github merge branches	Medium	Done	10		Elif
5	Creating relations between database tables	Medium	In Progress	12		Merve
6	Switch to log-in page from homepage	Medium	Done	12		Esra
7	Switch and add need page from homepage after successfull entry	Medium	Done	12		Esra
8	Add buttnos to track number of needs	Medium	Done	12		Esra
9	Pull need posts from database and publish them on homepage	Medium	Done	12		Esra
10	Correct the active need numbers on the homepage	High	Done	13		Esra and Medine
11	Add relation to need post and the user (username)	Medium	In Progress			
12	User must enter with their username instead of their name	High	Done	13		Merve
13	Error related to the NeedPostInformations class	Medium	Done	14		Esra
14	Showing detailed post information	Medium	Done	14		Merve

4. UML DIAGRAMS

4.1 CONTEXT MODEL

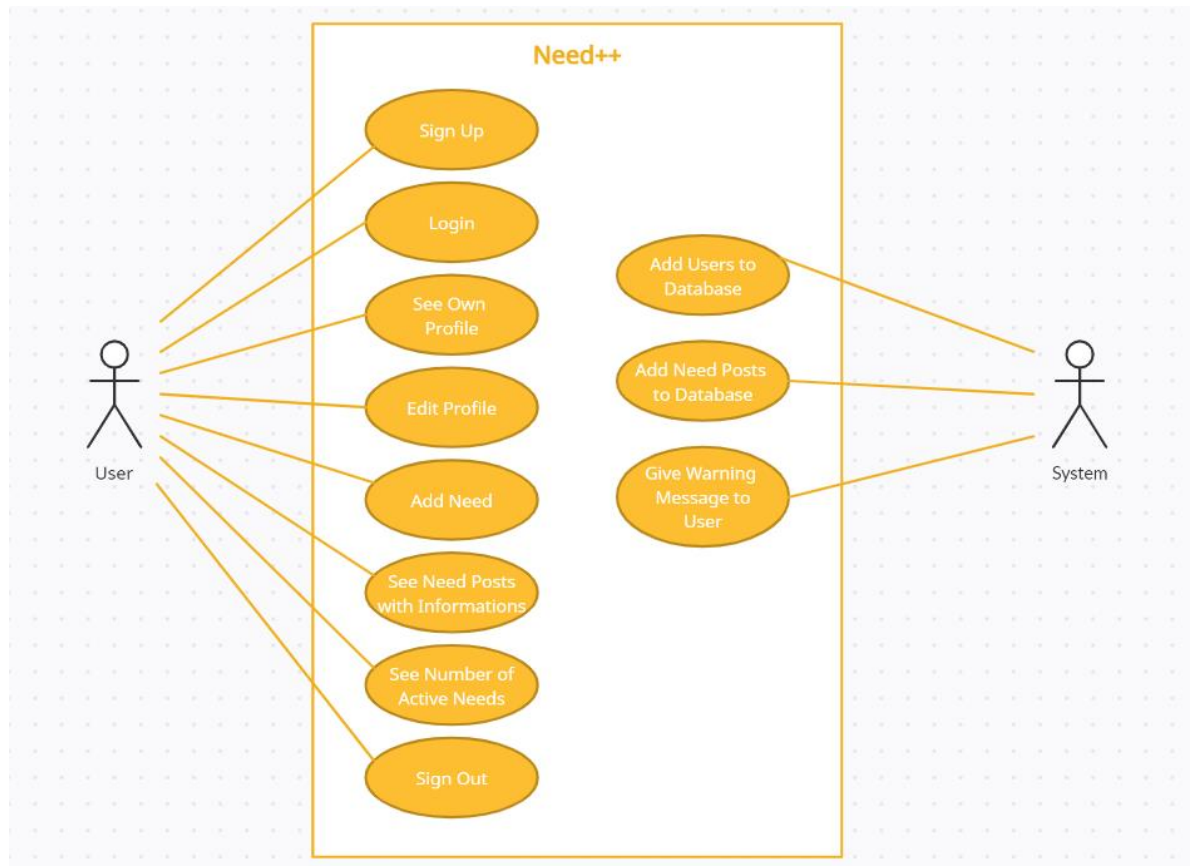


4.2 PROCESS MODEL



4.3 USE CASE DIAGRAMS and Its RELATED TABLES

Use-Case Diagrams



Note: Sign Out case added and text edits were made in some cases

Related Tables

Use Case Description - Sign Up

SYSTEM	Need++
USE CASE	Sign Up
ACTORS	User(needy or supplier)
DATA	System requests the necessary informations from the user to sign up. These informations are name, surname, birthday, email, username, password, country, city and district informations. These informations keeps in the database by the system.
STIMULUS	When the user clicks the sign up button and required informations is requested.
RESPONSE	The user enters the required informations. This informations are registered in the database. Then the user can use the system.
COMMENTS	The user must enter the information required to sign up in the system correctly. Otherwise, a warning message is shown. For example user should enter valid email.

Use Case Description - Login

SYSTEM	Need++
USE CASE	Login
ACTORS	User(needy or supplier)
DATA	User enters the necessary informations. These are username and password informations.
STIMULUS	The data entered by the user is compared with the datas in the database.
RESPONSE	If the datas match, the user successfully logs into the system. If there is no match, the system shows an information message to use user for checking its informations.
COMMENTS	The user must enter his/her informations correctly to log into the system. Otherwise, he/she cannot login to the system. For example, a warning message will be shown to the user in case of entering the password incorrectly.

Use Case Description - See Own Profile

SYSTEM	Need++
USE CASE	See Own Profile
ACTORS	User(needy or supplier)
DATA	On the profile page, there are informations recorded in the database that the user entered while registering in the system. These datas are name, surname, birthday, profile photo, email, username, password, country, city, district informations.
STIMULUS	When the user clicks the profile page button, informations are taken from the database.
RESPONSE	The informations taken from the database is displayed to the user.

Use Case Description - Edit Profile

SYSTEM	Need++
USE CASE	Edit Profile
ACTORS	User(needy or supplier)
DATA	User edits his/her profile informations. These are name, surname, birthday, profile photo, email, username, password, country, city, district.
STIMULUS	The user clicks on the profile page and clicks on the edit box next to the information he/she wants to change. Then the user is prompted to enter new data.
RESPONSE	The new information entered by the user is made current information.
COMMENTS	The user must enter the information as desired while editing his/her profile. For example, if the extension that is entered except the required extensions while editing the profile photo, it will not be accepted.

Use Case Description - Add Need

SYSTEM	Need++
USE CASE	Add Need
ACTORS	User(needy or supplier)
DATA	System requests the necessary informations from the user to share need post. These informations are title of need, need location, need definition informations. These informations keeps in the database by the system.
STIMULUS	When the user clicks the add need button and required informations is requested.
RESPONSE	The user enters the required informations. This informations are registered in the database. And the need post becomes visible on the homepage.
COMMENTS	The required informations should be entered as required. Otherwise, they cannot be added successfully.

Use Case Description - See Need Posts

With Informations

SYSTEM	Need++
USE CASE	See Need Posts with Informations
ACTORS	User(needy or supplier)
DATA	There is a table on the homepage that allows viewing all need posts. There are need title, number of need, need definition, need location and email informations about need post in this table.
STIMULUS	The user views the homepage.
RESPONSE	Need post's informations comes from the database to the table on the homepage, and user sees the all need posts.

Use Case Description - See Number of

Active Needs

SYSTEM	Need++
USE CASE	See Number of Active Needs
ACTORS	User(needy or supplier)
DATA	There is a box on the homepage that allows viewing number of active needs. There is number of active needs information in this box.
STIMULUS	The user views the homepage.
RESPONSE	Number of active needs comes from the database to the box on the homepage, and user sees the number of active needs.

Use Case Description – Sign Out

SYSTEM	Need++
USE CASE	Sign Out
ACTORS	User(needy or supplier)
DATA	No data.
STIMULUS	User clicks the sign out button.
RESPONSE	The user successfully exits the system.

Use Case Description - Add Users to
Database

SYSTEM	Need++
USE CASE	Add Users to Database
ACTORS	System
DATA	When the user sign up to the system succesfully his/her name, surname, birthday, email, username, password, country, city, district informations are added to the database by the system.
STIMULUS	The user enters the necessary information to sign up in the system.
RESPONSE	Users are added to the database with their informations.

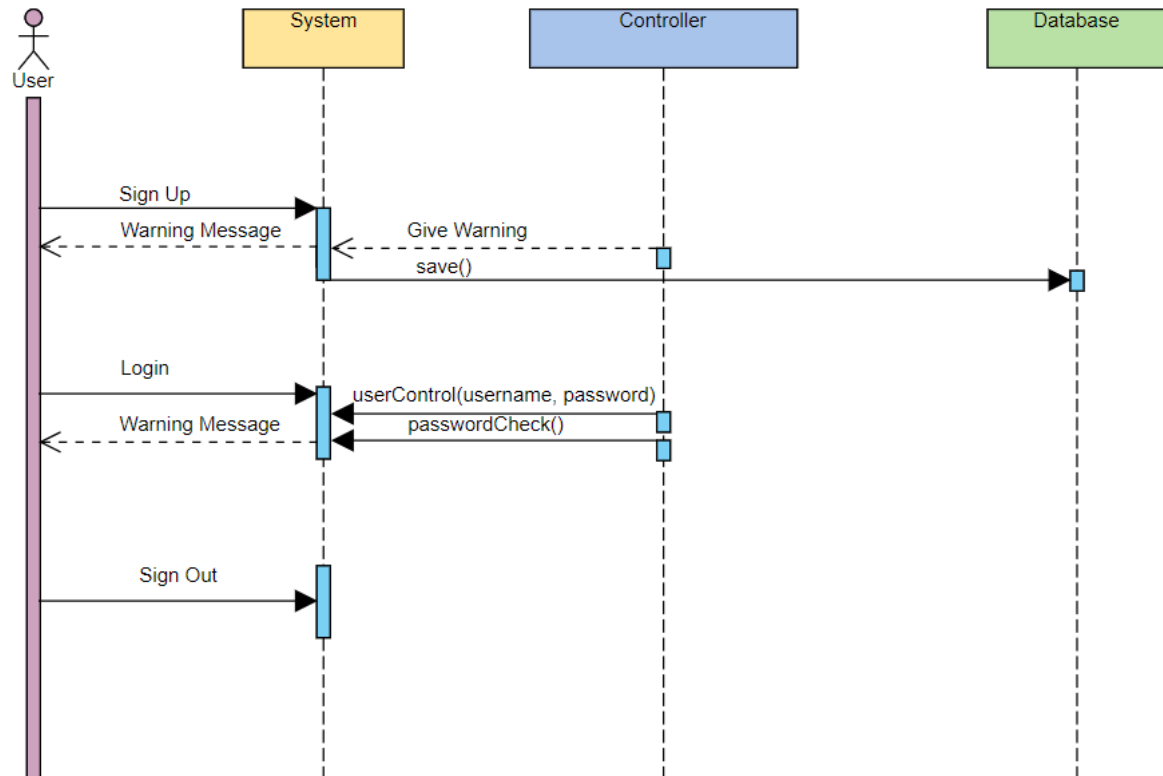
Use Case Description - Add Need Posts
to Database

SYSTEM	Need++
USE CASE	Add Need Posts to Database
ACTORS	System
DATA	When the user add need post to the system succesfully, title of need, need location, need definition informations are added to the database by the system.
STIMULUS	The user enters the necessary information to add need post in the system.
RESPONSE	Need posts are added to the database with their informations.

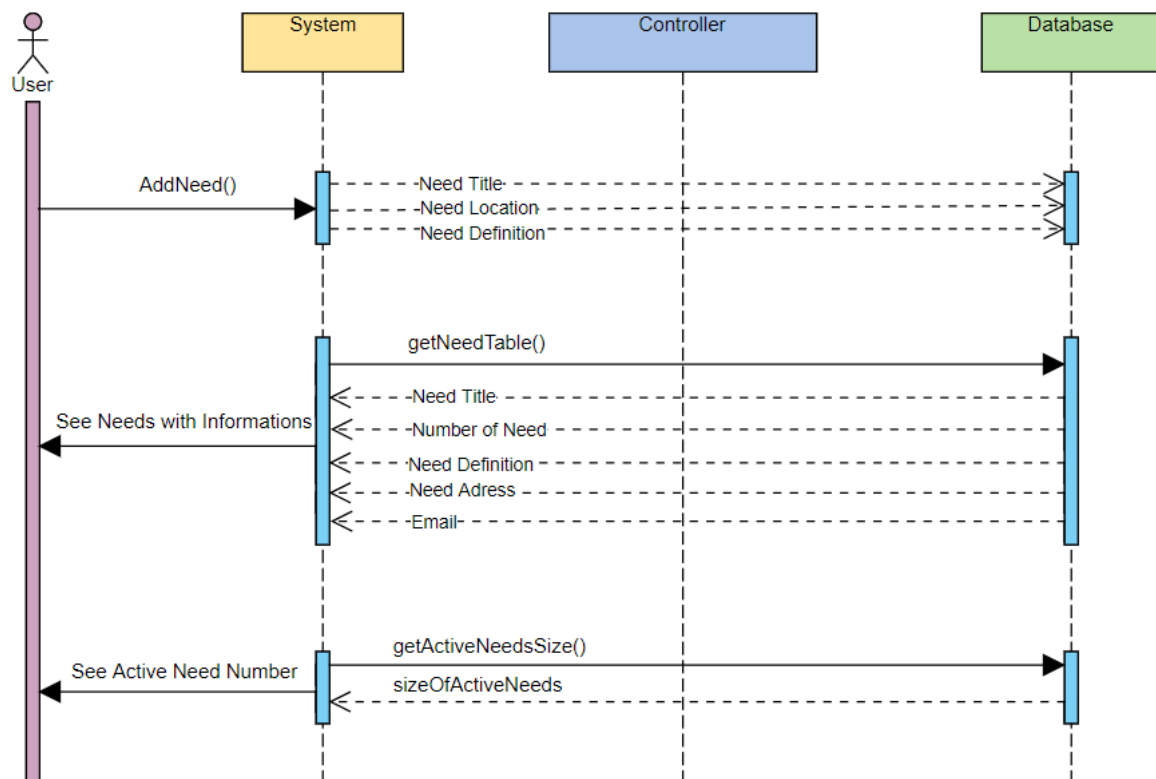
Use Case Description - Give Warning Message
to User

SYSTEM	Need++
USE CASE	Give Warning Message to User
ACTORS	System
DATA	The datas are checked for the action that the user wants to perform.
STIMULUS	Verification of the datas does not give successful results.
RESPONSE	A warning message is given to the user by the system.
COMMENTS	Warning messages to the user may occur in different situations. Such as wrong password, invalid birthday information.

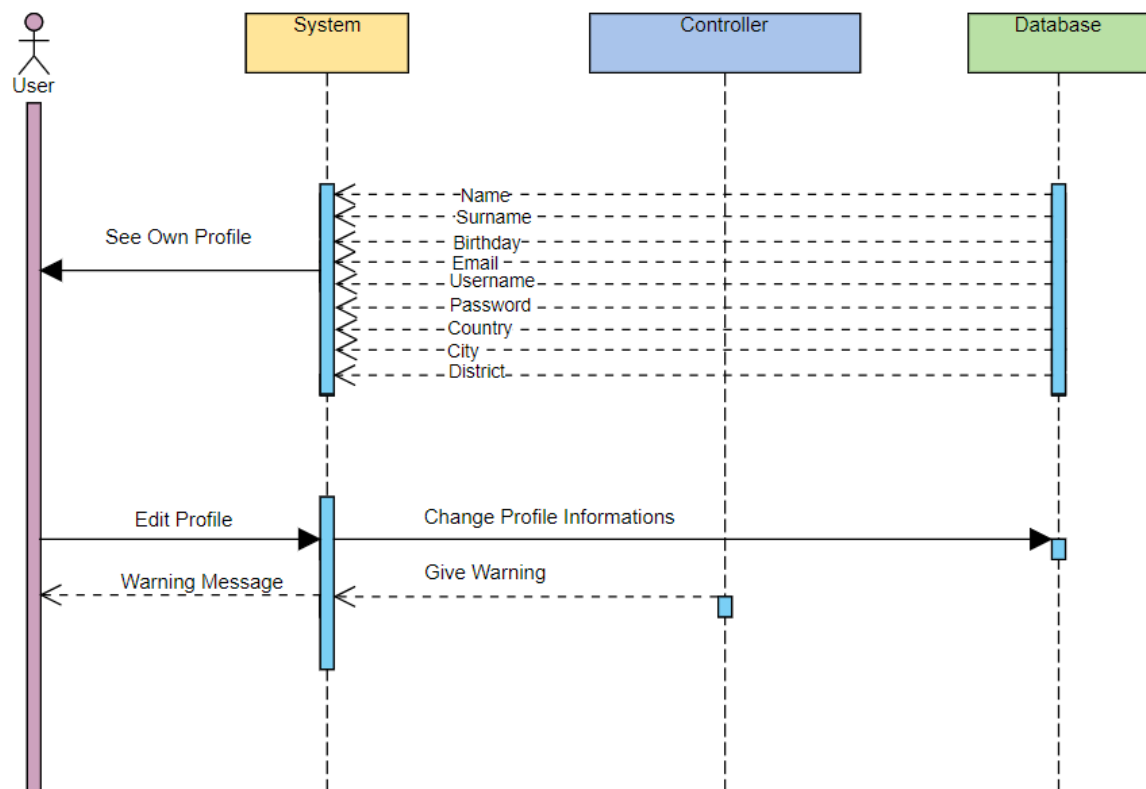
4.4 SEQUENCE DIAGRAMS



Sequence Diagram For Sign up, Log in and Sign out

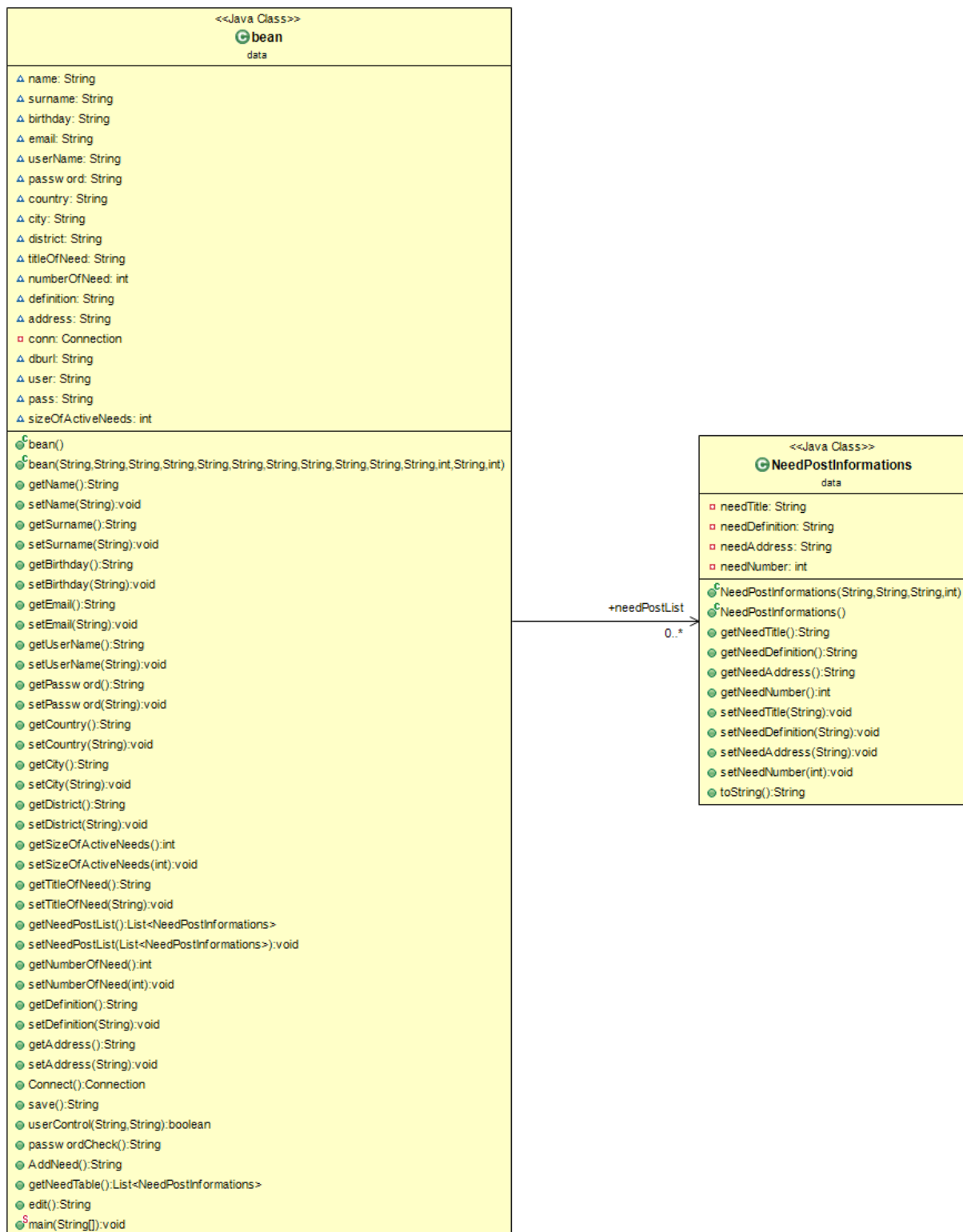


Sequence Diagram For Add Need, See Needs with Informations and See Active Needs Number

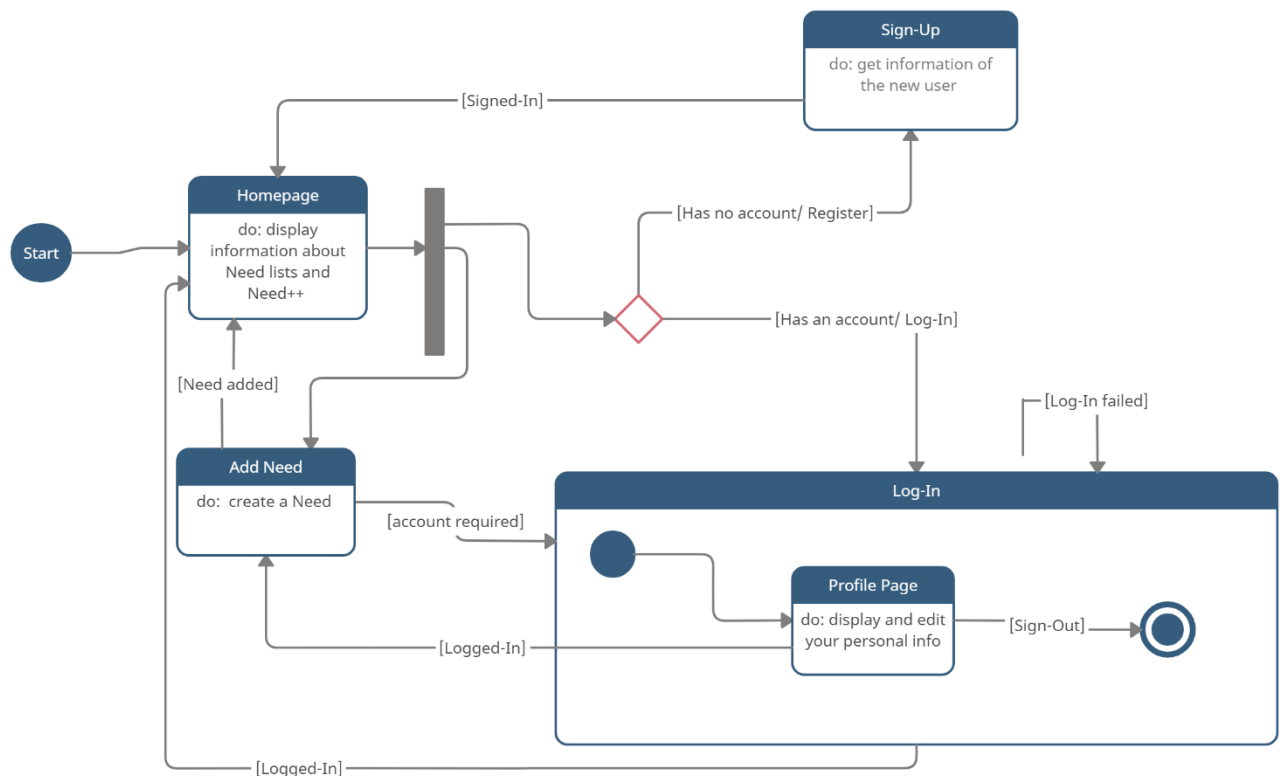


Sequence Diagram For See Own Profile and Edit Profile

4.5 CLASS DIAGRAM



4.6 STATE DIAGRAMS



5. SYSTEM ARCHITECTURE DOCUMENT

The system architecture is the conceptual model that defines structure, behavior, and more views of a system. Description of the system is the representation of the structures and the behavior of the system.

Project architecture consists of elements, such as software and hardware architectures, relationships between components, documentations, people's roles on the system etc...

As a java web application, Need++ system architecture uses the MVC design pattern model.

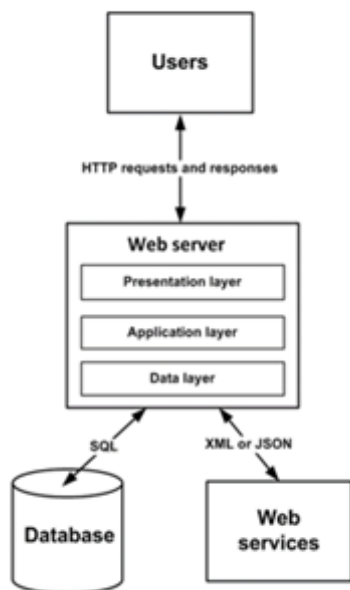
MVC is an architectural software design pattern, which is mostly used for developing user interfaces. MVC model separates the architecture into three parts: Model, View, Controller.

- **Model:** Manages the data, logic and rules of the application. Responsible for managing the data of the application. It receives user input from the controller.
- **View:** Any representation of the information.
- **Controller:** Accepts input and converts it to commands for the model or view. The controller responds to the user input and performs interactions on the data model objects.

Some Advantages of the MVC Model:

- Faster and easier development process
- Ability to provide multiple views
- The modification does not affect the entire model
- Code reusability

We choose to use this design pattern because the advantages in terms of web applications are considerably greater than the disadvantages.



Users: Users make requests via Need++ web page, and web server. Then receive responses using JSP (Java Server Pages). Then they Sign Up, Log In, Sign In...

Web Server: Hosts the application's various layers which conforms MVC:

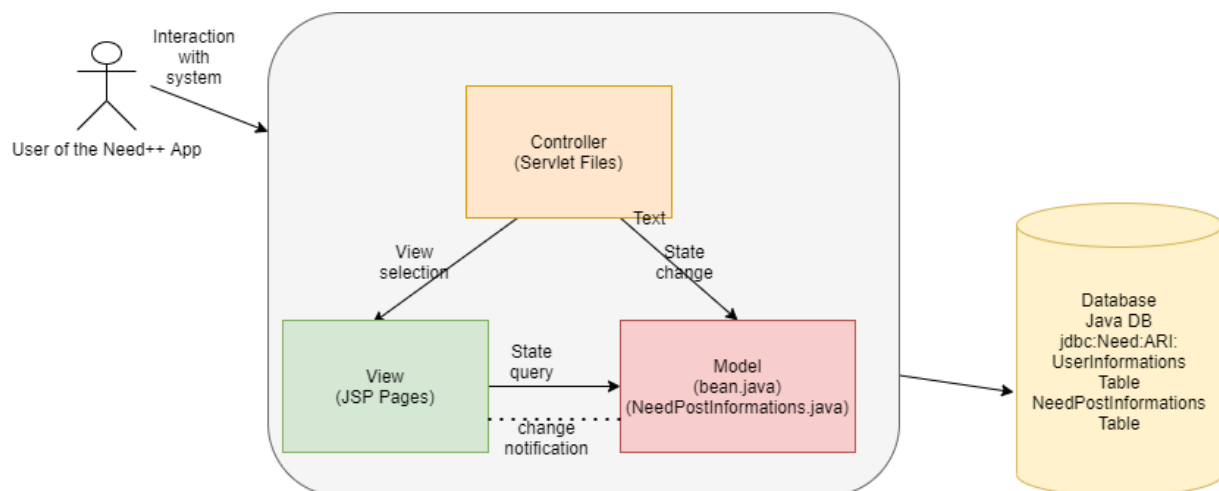
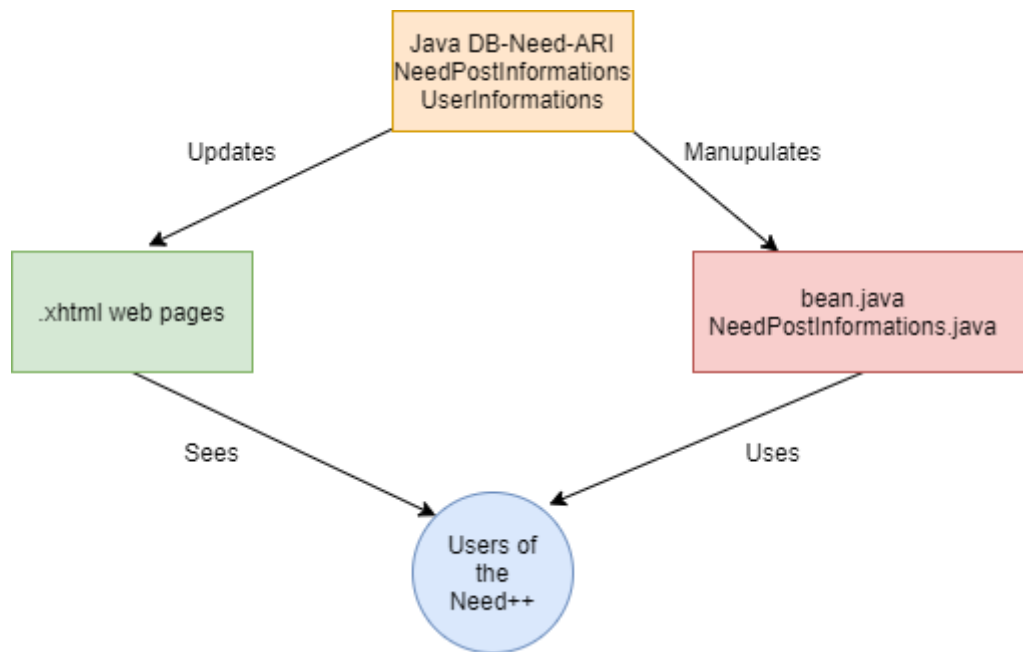
Presentation Layer: Users interact with application via HTTP requests and responses rendered in a browser.

Application Layer: Manages the flow of the application, implements the business logic and liaises with the data layer to process requests from users and their responses. Open-source, third party products reside here.

Data Layer: Handles domain data and provides persistence and retrieval services for the database.

Database: Where data is persisted and retrieved. Need++ has a JavaDB to keep the information of the users, and the need posts that users make.

Web Services: Interaction with other applications.



6. WORKING GRAPHICAL INTERFACE AND BACKEND SERVICES

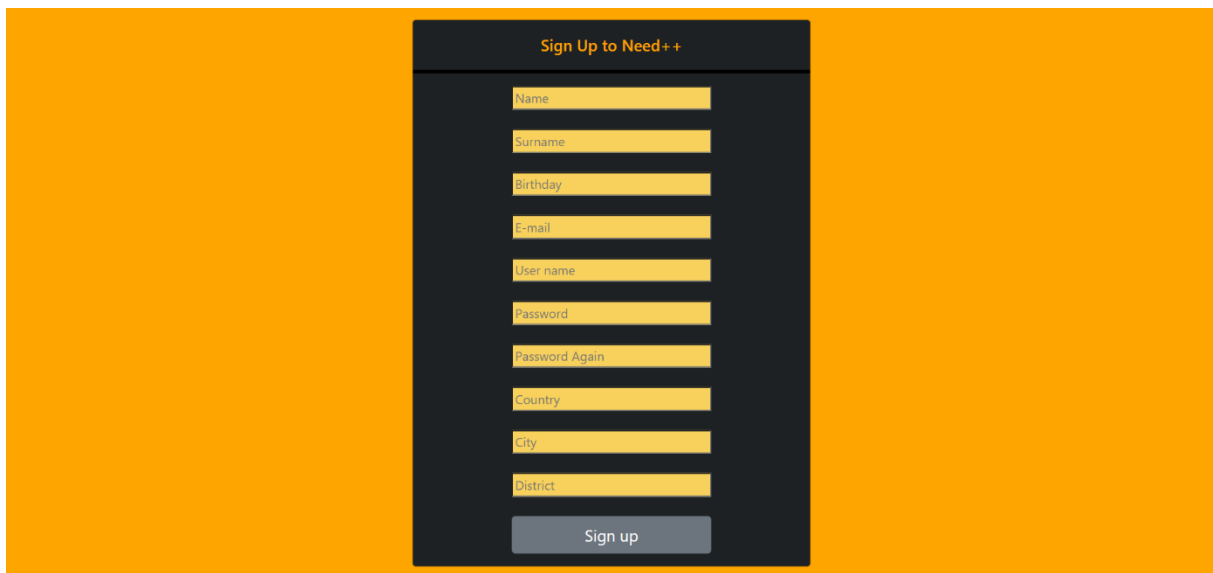
Gui and Database

A simulation was created in which 3 people were registered and total of 6 needs were entered to show the operation of the Gui and the database. Below are pictures of what we think is sufficient of these steps. (They can be unordered)



The screenshot shows the Need++ web application interface. At the top, there is a dark blue header with the Need++ logo on the left and "Log In" and "Sign Up" buttons on the right. The main content area has an orange background. On the left, there is a table with the following columns: Need Title, Number of need, Need Definition, Need Address, Need Username, and Redirect. The table contains six rows of data. To the right of the table, there is a box labeled "15 Active Needs" and a button labeled "Add Need".

Need Title	Number of need	Need Definition	Need Address	Need Username	Redirect
pencil	20	Pen for my child	Bilecik	abab	View details
Paper	100	Paper for my students	Çanakkale	eu	View details
Boots	15	Winter boots for students	Van	be	View details
Reading books	48	Reading books for students	Antalya	merve	View details
Guitar	2	Guitar for our course	Samsun	ersfvg	View details
Wheelchair for					View details



The screenshot shows the "Sign Up to Need++" form. The form is a dark blue box with a title "Sign Up to Need++" at the top. It contains several input fields for user registration: Name, Surname, Birthday, E-mail, User name, Password, Password Again, Country, City, and District. At the bottom of the form is a "Sign up" button.

Sign Up to Need++

Ufuk

Yaver

28.01.1986

ufuk_yaver@gmail.com

ufuk1905

1905

1905

Turkey

Izmir

Dikiİ

Sign up

Log in to Need++

Username

Password

Forgot password?

Log in

Don't have an account? Sign up

Log in to Need++

ufuk1905

1905

Forgot password?

Log in

Don't have an account? Sign up

Add Need

Baby Diaper

Mecidiyekoy/Istanbul

Baby diaper for 6 months

Add need

About Us

Hello we are Team An. Need ++ was designed to be a social aid themed web application and we continue to develop our project.

The web application called Need ++, which we plan to design within the scope of our software engineering course, is an online platform that we plan to bring together the needy and benefactors without anyone in between.

With this web application, individual or collective assistance requests can be made and people who want to meet them can deliver their help to those in need.

HELP

How To Use?

You can see all the published needs on the home page, "view detailed" shows information about need and needy. If you want to add a need; using "Add need" button after logging in if you have an account; If you do not have an account, you can add a need by using the "Add need" button after creating an account. If you want to view and edit your profile, click the Profile button.

Managing Your Account

If you want to create a new account, you can register by clicking the "Sign up" button and filling in the required information. If you have an account, you can log in using the "Log in" button; You can log out by using the "Sign out" button. If you want to edit your profile, you can click the "Profile" button, then the "Edit Profile" button and update whatever you want!


Safety and Security

Law No. 6698 on the Protection of Personal Data ("KVKK") and other relevant legislation page, as Need ++, as Data Supervisor, within the framework of disclosing your personal information; We would like to inform you, the users, that it will be recorded, stored, updated, disclosed / transferred to third parties to be informed of the legislation, classified and processed according to other procedures and principles specified in the relevant legislation.

Rules and Policy

Ownership of the Site: The site is owned and controlled by the Need ++ team. Access to Password Protected / Secure Areas: Access to password protected and / or secure areas of the site is only available to authorized users. Legal proceedings may be initiated against unauthorized individuals attempting to enter these areas of the site. Jurisdiction / Judge Law: This site's users are agreeing to abide by the laws REPUBLIC OF TURKEY.

Edit Profile



Welcome ysf

To be able to edit, please fill in all the boxes below

Yusuf

Dilber

06.06.1996

ysf@gmail.com

ysf123ysf

Turkey

Ist

Pendik

Edit

[illegible]

There are three result cases:

- 1- Success,
- 2- Not success,
- 3- and Block (not eligible for testing)

The errors that generated in the first couple stages are eliminated. And the success rate reached to %100 at the end. (for non-block tests)

- Total number of tests done: 42
- Total number of tests passed: 42
- Total number of tests failed: 0

Deatiled Test Results

The table below shows the detailed information of the tests done, such as input values, expected and final outputs.

no	Test Case Name	Behavior	Expected Output	Test Data/Input	Result	Fail	Notes
1	getName:	returns the name of the user	the name of the user	elif	Success		
2	setName	changes the name	new name set to EsraNur	EsraNur	Success		
3	getSurname	returns the surname of the user	keles	keles	Success		
4	setSurname	changes the surname	new surname set to Arican	Arican	Success		
5	getBirthday	returns the birthday	01.01.1911	01.01.1901	Success		
6	setBirthday	changes the birthday	birthday updated to 11.11.1911	#####	Success		
7	getEmail	returns the eMail	elif@hotmail.com	elif@hotmail.com	Success		
8	setEmail	changes the eMail	email updated	newmail@hotmail.com	Success		
9	getUserName	returns the username	elif1	elif1	Success		
10	setUserName	changes the username	userName updated as newUserName	newUserName	Success		
11	getPassword	returns the password	pass123	pass123	Success		
12	setPassword	changes the password	password changed	changingPassw	Success		
13	getCountry	returns the country	tr	tr	Success		
14	setCountry	changes the country	country updated as türkiye	türkiye	Success		
15	getCity	returns the city	kocaeli	kocaeli	Success		

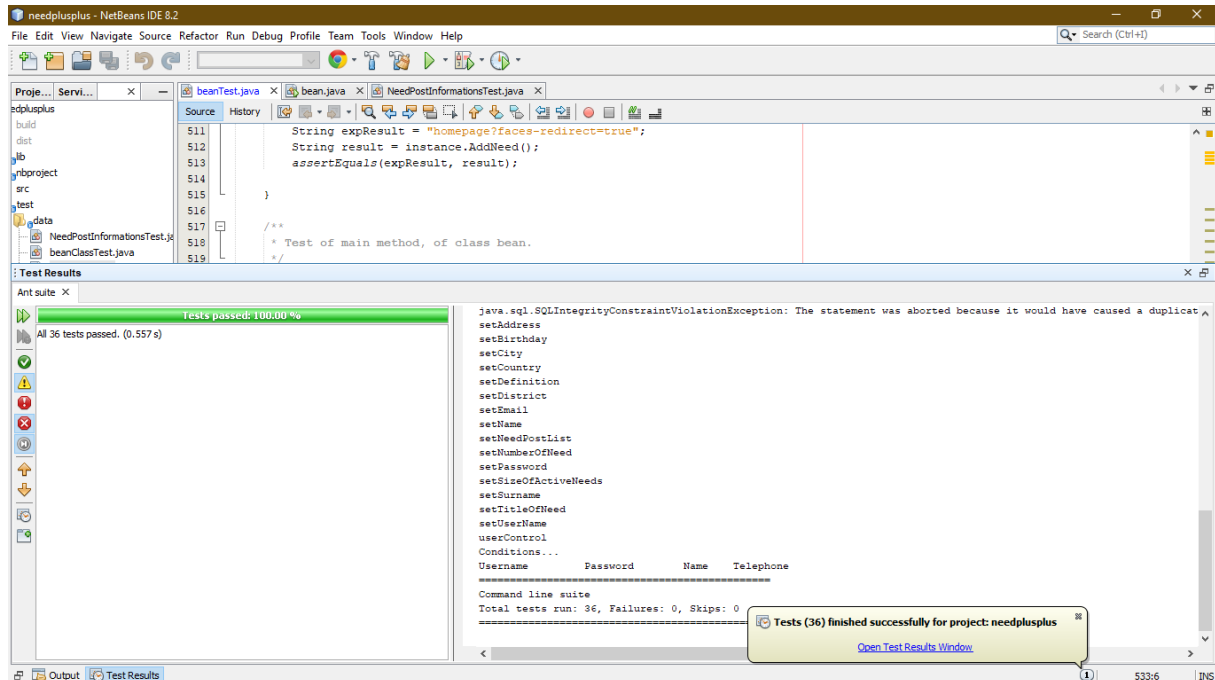
16	setCity	changes the city	city name updated as Muş	Muş	Success	
17	getDistrict	returns the district	gebze	gebze	Success	
18	setDistrict	changes the district	district updated to malazgirt	malazgirt	Success	
19	getSizeOfActiveNeeds	returns the number of active needs	0	0	Success	
20	setSizeOfActiveNeeds	changes the number of active needs	number changed to 1	1	Success	
21	getTitleOfNeed	returns the title of the need	Needtitle	Needtitle	Success	
22	setTitleOfNeed	changes the title of need	new title set as aNewTitleForNeed	aNewTitleForNeed	Success	
23	getNeedPostList				Block	
24	setNeedPostList				Block	
25	getNumberOfNeed	returns the number of need	5	5	Success	
26	setNumberOfNeed	changes the number of need	new number set as 10	10	Success	
27	getDefinition	returns the definition of need	SimpleDefinition	SimpleDefinition	Success	
28	setDefinition	changes the definition of need	definition updated	aNewDefinitionForNeed	Success	
29	getAddress	returns the address	simpleAddress	simpleAddress	Success	
30	setAddress	changes the address	address updated to newAddress	newAddress	Success	
31	connect				Block	connection class returns a different code in each function call
32	save	checks if the saving is successful	homepage?faces-redirect=true		Success	Directs to homepage if save is successful.
33	userControl	checks if user entered their info correct			Success	
34	passwordCheck	checks if the password is correct	homepage?faces-redirect=true		Success	Directs user to homepage if the password is correct. If not the wrong password page comes up
35	addNeed	adds a new need	homepage?faces-redirect=true		Success	

36	main				Block		
37	getNeedTittle	returns the title of need post	Needtitle	Needtitle	Success		
38	getNeedDefinition	returns the definion of the post	SimpleDefinition	SimpleDefinition	Success		
39	getNeedAddress	returns the address of the post	simpleAddress	simpleAddress	Success		
40	getNeedNumber	returns the number of the needs of post	5	5	Success		
41	setNeedTittle	sets a new title to the post	a new title set	newTitle	Success		
42	setNeedDefinition	changes the definition of the post	definition updated	newDefinition	Success		
43	setNeedAddress	updates the address	address changed	newAddress	Success		
44	setNeedNumber	changes the need number	need number updated	10	Success		
45	toString	writes the parts in one string	strings are the same	bean info class' fields	Success		
46	edit	checks if the editing is successfull	homepage?faces-redirect=true		Success		Directs to homepage if edit is successful.

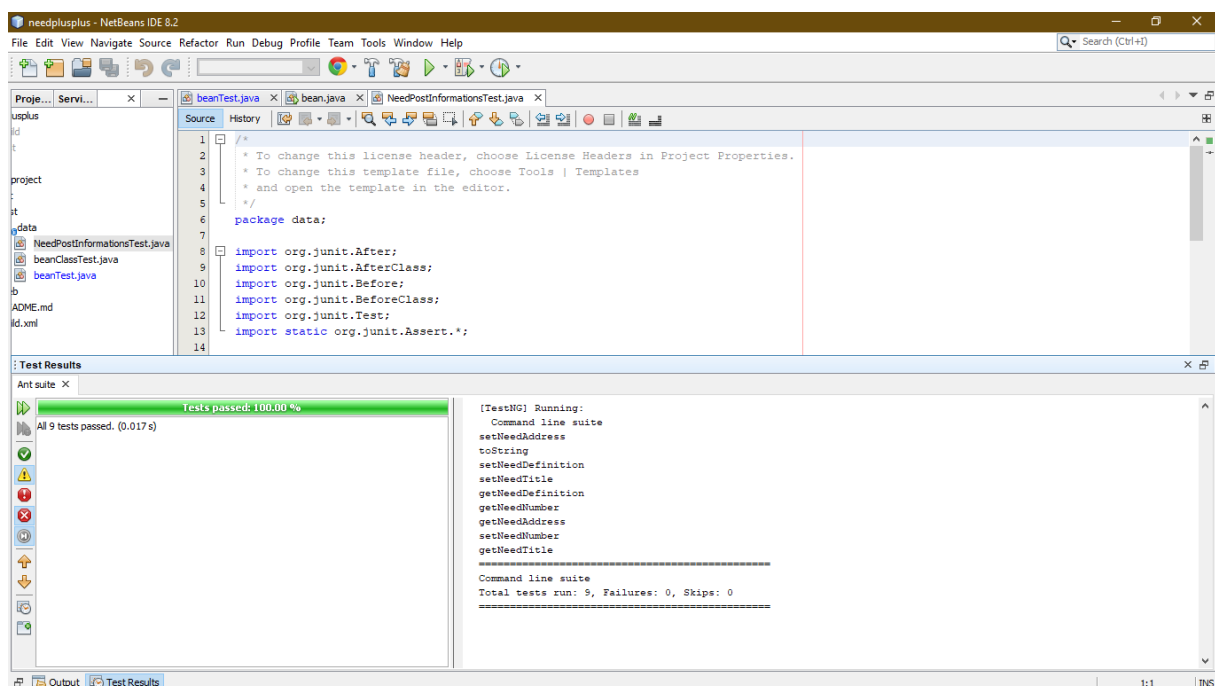
Note: The tests shown are also done with some wrong input values. And these wrong values produced the error message that the tests would end normally.

Screenshots For the Unit Tests

beanTest.java

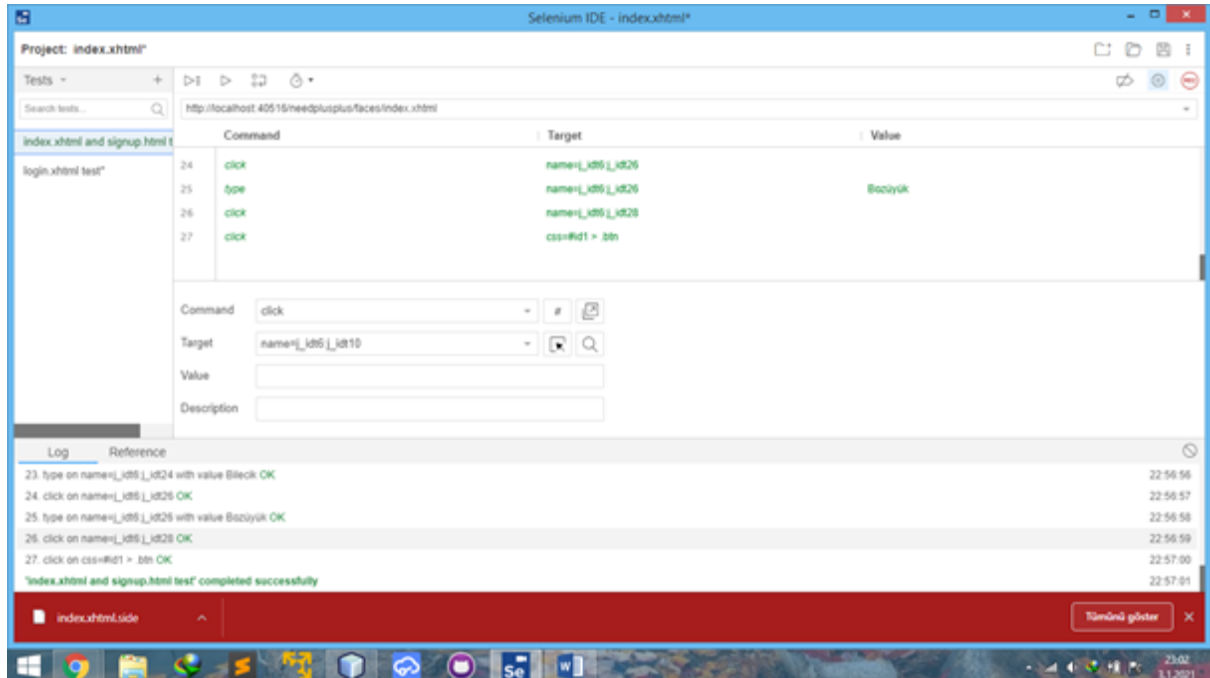


NeedPostInformationsTest.java

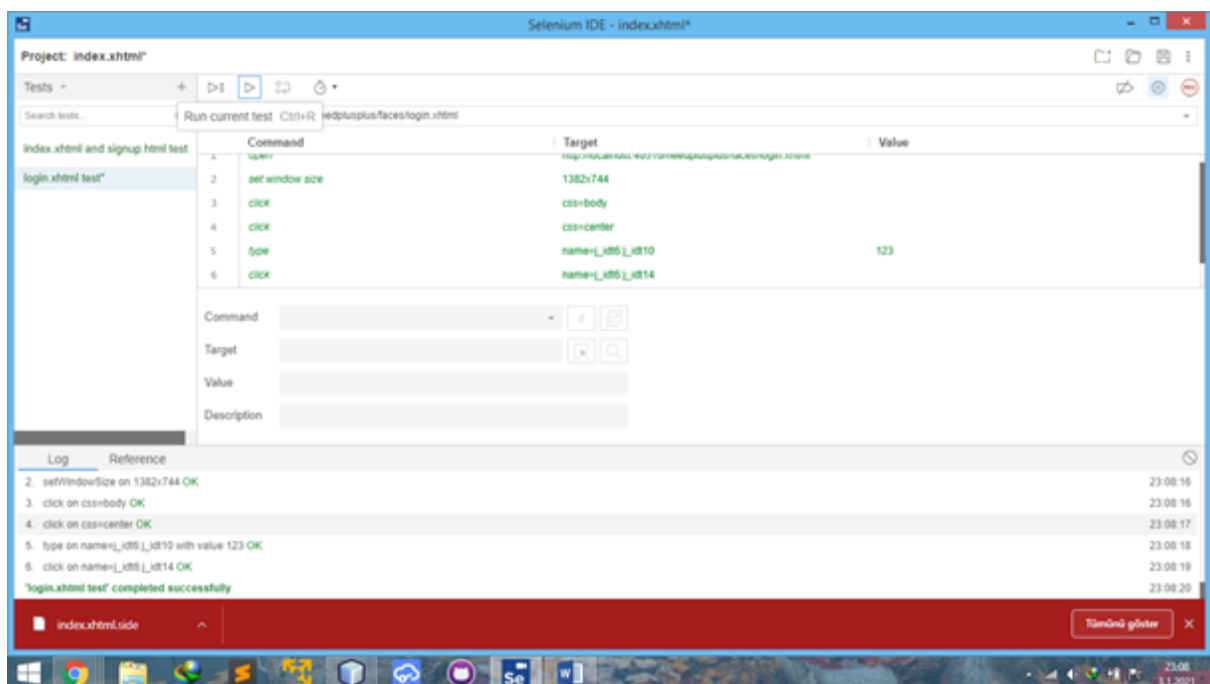


Screenshots For the Selenium Tests

index.html and signup.html test



login.xhtml test



addneed.xhtml test

The screenshot shows the Selenium IDE interface for a project named 'index.xhtml'. The test suite 'addneed.xhtml' is selected, and the 'Run current test' button is highlighted. The test suite contains five test cases: 'index.xhtml and signup.html test', 'login.xhtml test', and three individual test cases for 'addneed.xhtml' (steps 7, 8, and 9). The test results show that all tests passed successfully.

Command	Target	Value
type	name=_id6_j_id12	
click	name=_id6_j_id12	
type	name=_id6_j_id12	pencil 45
click	css=center	
type	name=_id6_j_id14	45
click	name=_id6_j_id16	

Log Reference

- 7. click on name=_id6_j_id12 OK 23:14:51
- 8. type on name=_id6_j_id12 with value pencil 45 OK 23:14:52
- 9. click on css=center OK 23:14:53
- 10. type on name=_id6_j_id14 with value 45 OK 23:14:54
- 11. click on name=_id6_j_id16 OK 23:14:55
- 'addneed.xhtml' completed successfully 23:14:56

Profile.xhtml test

The screenshot shows the Selenium IDE interface for a project named 'profile.xhtml'. The test suite 'Profile page' is selected, and the 'Run current test' button is highlighted. The test suite contains five test cases: 'Profile page', 'http://localhost:40516', and three individual test cases for 'Profile page' (steps 14, 15, and 16). The test results show that all tests passed successfully.

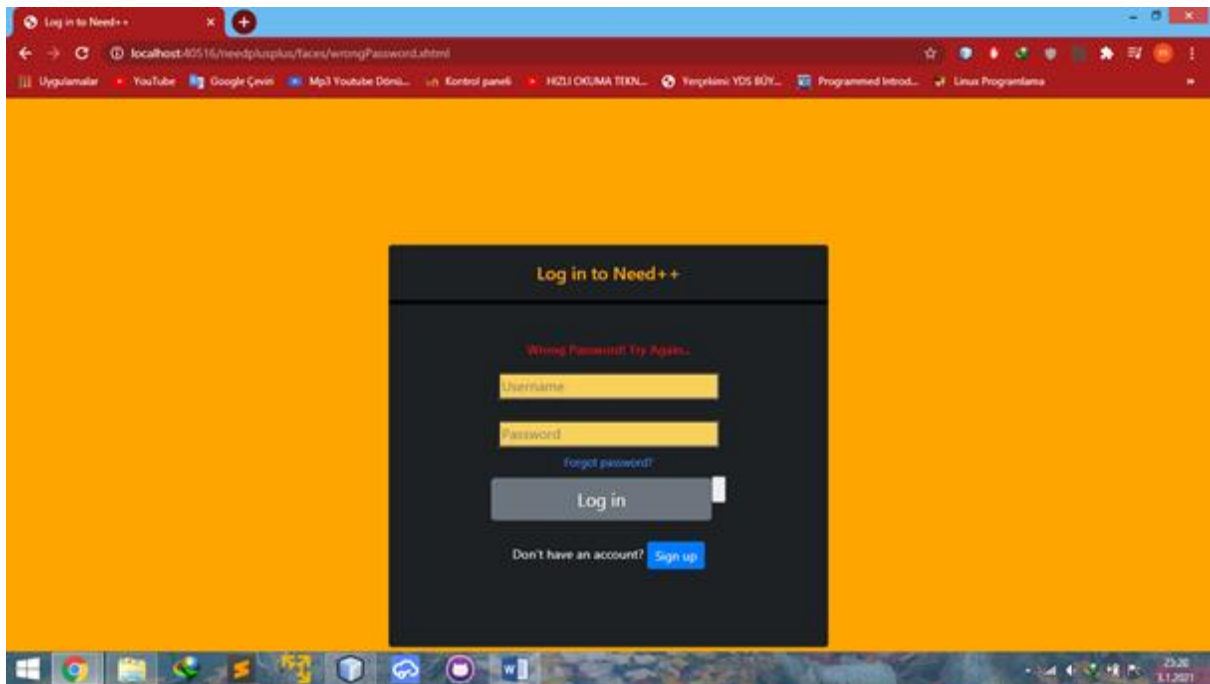
Command	Target	Value
click	css=center	
click	name=_id6_j_id22	
type	name=_id6_j_id22	Turkey
click	name=_id6_j_id24	
type	name=_id6_j_id24	er
click	name=_id6_j_id26	
type	name=_id6_j_id26	re
click	name=_id6_j_id28	

Runs: 1 Failures: 0

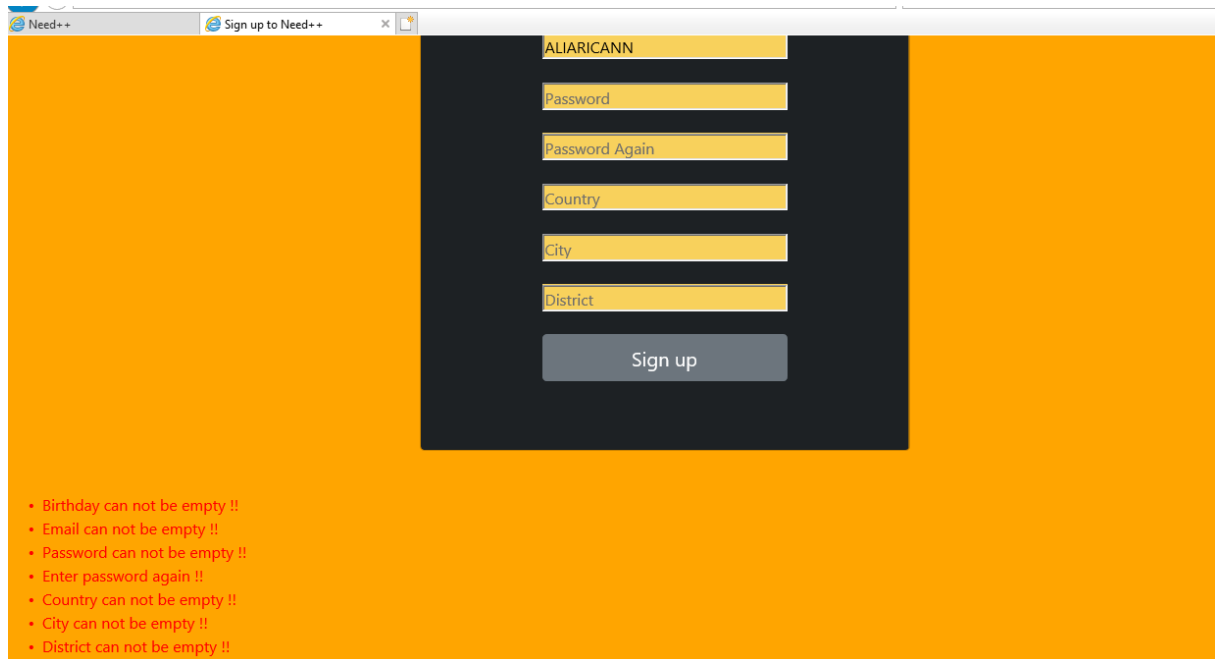
Log Reference

- 17. click on name=_id6_j_id24 OK 16:47:53
- 18. type on name=_id6_j_id24 with value er OK 16:47:53
- 19. click on name=_id6_j_id26 OK 16:47:53
- 20. type on name=_id6_j_id26 with value re OK 16:47:53
- 21. click on name=_id6_j_id28 OK 16:47:53
- 'Profile page' completed successfully 16:47:53

WrongPassword Control



Sign up can not pass blank control



Editing can not pass blank control

Welcome ysf

To be able to edit, please fill in all the boxes below

Name

Surname

Birthday

E-mail

ysf123ysf

Country

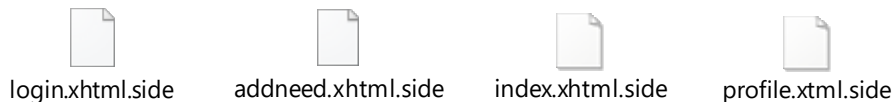
City

District

Edit

- Name can not be empty !!
- Surname can not be empty !!
- Birthday can not be empty !!

The selenium test files for some cases



8. NEED++ PROJECT REQUIREMENT ANALYSIS

8.1. PREFACE

This document has been written to explain how potential users of the system can use the software.

Version 1: written at 25.10.2020

Version 2: written at 20.11.2020. This version is more detailed than version 1.

Version 3: written at 28.12.2020. We have removed the estimated percentage improvement from our document because we thought we could not calculate them correctly.

8.2. INTRODUCTION and SCOPE OF THE PROJECT

The Need ++ project that we are developing as part of the software engineering course was designed as a web application. With this application, we aimed to develop an important

project in the name of social assistance by bringing people in need and who wants to help them together under the same platform.

With this application, those in need can share their help requests on various issues, and those who want to help them will have the opportunity to directly deliver their help to these people.

Basically, the main features planned to be included in our application are as follows:

- Declaration of need
- Need fulfillment

With the declaration of need feature, users can share anything that they need on this platform on any subject. For example, baby clothes, stationery, etc.

Users of the system can also support those in need with the need-solving feature, in matters that you think you can help from among the postings shared on the site.

8.3. FUNCTIONAL REQUIREMENTS

8.3.1 USER REQUIREMENTS

8.3.1.1 MEMBER

Member is the main target of the application. First anyone who wants to use this application should register first. Members informations such as name, surname, phone number, address are kept in database. But these personal informations are hidden from other users.

After registration members can share need posts on the web application. Any member can see shared need posts on the application and they can also support the needy.

- System should allow member to sign up the application.
- Member can sign in the application using necessary informations such as user name and password.
- System shall allow member to see his/her profile page.
- System should allow member to edit his/her profile page.
- System should allow user add profile photo.
- User can see all shared need posts.
- User can add new post for his/her needs.
- System should allow user to need fulfillment.
- System should allow user to see needy's information (if they need to contact each other).

8.3.2 SYSTEM REQUIREMENTS

- The system shall allow for online post sharing for all users.
- The system shall reflect a new and changed post description within x minutes of the database being updated by the program itself.
- The system shall display information that is customized based on the user's profile, and location . - The system shall allow users to view the owner of any post. A user should be able to contact the owner of post in given contact information.
- The system shall provide users who want fulfillment of someone's need with accurate needy's address and phone data (shipping information)
- The system shall provide a search facility that will allow full-text searching of all web pages that the user is permitted to access. The system must support the following searches:
 - find all words specified
 - find any word specified

This feature will save time by allowing the user to search for the need posts according to their address or category of the shared need.

- The system shall allow the user's status to be stored for the next time he/she returns to the web site.
- The application will provide users with different language options. In this way, we plan to improve customer service.
- System should control when new user is registering, she/he must enter same password twice.
- System should control given e-mail address is valid or not.
- When a password change request is made by the users, the system should keep the old password under record so that the user does not use one of their old passwords again.
- System should be able to connect with user via e-mail when user forgets his/her current password.
- System should check extension of profile photo (.jpg , .jpeg, .png)

8.4. NON-FUNCTIONAL REQUIREMENTS

8.4.1 Usability

- Users should be able to use the application immediately after registering.

- A user can not use the application in more than one browser tab at the same time.

8.4.2 Security

- The system shall provide password protected access to web pages that are to be viewed only by users
- Transaction data must be transmitted in encrypted form.

8.4.3 Supportability

- The system should be able to accommodate new need posts without major reengineering.
- After any modification, sharing post, new membership etc. database shall be synchronized.
- The system web site shall be viewable from Internet Explorer 4.0 or later, Google Chrome, and Firefox.

8.4.4 Online user Documentation and Help

- The system shall provide a web page that explains how to navigate the web application. This page should be customized based on what pages that user is allowed to access.
- This help page should be accessible from all other pages.

8.4.5 Purchased Components

- A language translation tool from English to Turkish will be needed.
- A web site search engine will be needed.

8.4.6 Interfaces

- Software's language is Java.
- Using programming technique: Object Oriented Programming.
- Used web application developing tool: Netbeans
- Used service Apache Tomcat 9.0.40

The system must interface with

- The current SQL database systems(Java DB)
- The acquired language translation tool
- The acquired web site search engine

9. CONTINUOUS INTEGRATION and CONTINUOUS DELIVERY CONCEPTS

CI/CD is a good practice to devops teams to implement. It is also an agile methodology best practice, as it enables software development teams to focus on meeting business requirements, code quality, and security because deployment steps are automated.

Continuous integration (CI) is a practice where a team of developers integrate their code early and often to the main branch or code repository. The goal is to reduce the risk of seeing “integration hell” by waiting for the end of a project to merge the work of all developers.

One of the primary benefits of adopting CI is that it will save time during development cycle by identifying and addressing conflicts early. It’s also a great way to reduce the amount of time spent on fixing bugs and regression by putting more emphasis on having a good test suite. Also, it helps share a better understanding of the codebase and the features that developed for potential users.

Automated testing- especially unit testing – is primarily key of CI. To get the full advantages of CI, team need to automate tests to be able to run for every change that is made to the main repository. Running tests on every branch of repository and not just focus on the main branch is more preferable. This technique helps to capture issues early and minimise disruptions for the team.

While automating tests is a key part of CI, it is not enough by itself. Developers on the team have to be careful about not to work for days on a feature without merging their changes back to the main branch.

It is important that developers integrate their changes as soon as possible on the main repository. If you let the code sitting on a branch or the developer workstation for too long, then we expose ourselves to the risk of having too many conflicts to look at when deciding to merge things back to the main branch.

By integrating early, you reduce the scope of the changes which makes it easier to understand conflicts when you have them. The other advantage is to make it easier to share knowledge among developers as they will get more digestible changes.

We can summarize the main steps behind continuous integration as follows:

1. Writing tests for the critical parts of codebase.
2. Getting a CI service to run those tests automatically on every push to the main repository.
3. Integrating changes everyday.
4. Fix the build as soon as it's broken.
5. Writing tests for every new story that implemented.

Continuous delivery or CD for short is a collection of many prior successful agile and organizational best practices. CD focuses an organization on building a streamlined, automated software release process. At the heart of the release process is an iterative feedback loop. The feedback loop revolves around delivery of software to the end user as quickly as possible, learning from their hands-on experience, and then incorporating that feedback into the next release.

Continuous delivery picks up where continuous integration ends. CD automates the delivery of applications to selected infrastructure environments. Most teams work with multiple environments other than the production, such as development and testing environments, and CD ensures there is an automated way to push code changes to them.

Continuous delivery deploys CI-validated code to the application.

All code is tested throughout each stage to secure better quality builds and applications with fewer bugs. CI/CD pipelines can determine what happens when builds pass or fail these tests, meaning that errors are identified much faster. As code goes through each stage of the development process, it's continually validated against many other changes in the repository happening concurrently, which ensures code integrity throughout the pipeline. Together, CI and CD accelerate how quickly your team delivers results for customers.

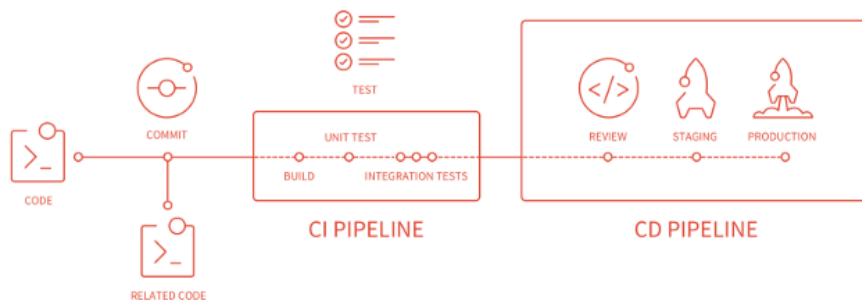


Figure 9.1 CI/CD Concepts Visualization



As a team working in different environments and remotely, we think that the CI principle is applicable in our project, especially due to the situations that everyone in the team works with different modules in the project.

While one person in the team works on the view part, another person works on the controls or on the database, in this case, applying CI in order not to encounter problems in the coding phase of our project, will help the development of the project with less problems.

In this context, since the day we opened our project repository, we have tried to apply the CI principle by dividing the work to be done into issues and using these issues in the branches in order to prevent conflicts. In the 1st and 2nd prototypes of the project, we chose to work on different branches and merge these prototypes with the main branch only on the dates we will release. Under the main branch, develop branch includes all the work on coding and the important work that is expected to be done before each prototype under this branch is divided into separate branches.

Another issue, which is one of the most important items of the CI principle, is unit testing, and the tests should be given importance to prevent other modules from being affected by the changes made. In order to implement this, we did not forget to test each new module we wrote as a team. To give an example, the feature of showing need post informations on homepage was not shown in the frontend, without testing the functioning of the written functions. If we tried to show this property in the view without making sure that this method works correctly, we could not understand where our error originated and we could run into a deadlock in the project. We tried to test all our modules like this example. None of the functions written were added to the project without being tested in itself, so the project continued to work as a whole with the new features.

While applying these principles, we have just discovered tools such as Jenkins, GitLab, Bamboo that provide CI / CD services, since we did not have a good knowledge of these principles when we started our project, and therefore we would like to state that we could not use them in our project.

Overview	Yours	Active	Stale	All branches	Search branches...
All branches					
main	Updated last month by EsraNurArican	Default			
dev_issue_11	Updated 43 minutes ago by ElifKeles	3 58		New pull request	
Selenium-Tests	Updated 27 days ago by MerveDur	0 1		New pull request	
Develop	Updated last month by EsraNurArican	3 37		New pull request	
Develop_issue_11	Updated last month by EsraNurArican	2 0		#15 Merged	
Develop_issue_8	Updated last month by EsraNurArican	41 1		New pull request	
Develop_Tssue_6	Updated 2 months ago by EsraNurArican	62 0		#7 Merged	
Develop_Tssue_2	Updated 2 months ago by EsraNurArican	66 0		New pull request	
Develop_Tssue_1	Updated 2 months ago by EsraNurArican	66 0		New pull request	

Figure 9.2 Branches on Github Repository of the Need++ Project

<input type="checkbox"/>	0 Open	✓ 6 Closed
<input type="checkbox"/>	Selenium Test	#17 by MerveDur was closed 5 minutes ago
<input type="checkbox"/>	Database Errors bug	#11 by EsraNurArican was closed 5 minutes ago
<input type="checkbox"/>	Week 8 Sprint Tasks	#8 by EsraNurArican was closed 5 minutes ago
<input type="checkbox"/>	Version_0_First_Try wontfix	#6 by EsraNurArican was closed 5 minutes ago
<input type="checkbox"/>	Test Frontend enhancement	#2 by EsraNurArican was closed 5 minutes ago
<input type="checkbox"/>	Frontend html pages will be added	#1 by EsraNurArican was closed 5 minutes ago

Figure 9.3 Closed Issues on repository to relate with branches

10. ACCESIBILITY OF THE PROJECT TO THE POTENTIAL USERS

10.1. Domain and Hosting, Transfer of Files from Local Area, Usability and Accesibility

After registration, domain in the form of "needplusplus.cf" was received from my.freenom.com. Then, to get hosting, "infinityfree.net" was registered first and then used. The required information was entered during these operations and all adjustments were completed. Domain and hosting accounts are linked to each other via nameservers. FileZilla was downloaded as an FTP tool program to transfer our files that we have already prepared and are located locally. FTP information (username, password and hostname) received from infinityfree.net is placed in FileZilla as required. Our files in local have been moved to the folder named htdocs. But after doing this at the first stage, although we have an index file, "Directory listing" was displayed on our website. It turned out that the reason is that our index file is "index.xhtml" but there is no index.xhtml for DirectoryIndex in the .htaccess file. That's why a new .htaccess file with index.xhtml was created. After doing this, the problem of "Directory listing" disappeared, but although all our files were transferred, only "index.xhtml" was displayed on the site. Efforts have been made to resolve this issue that occurred during the file transfer phase, but it could not be fixed. The picture is as shown below and you can view it from "needplusplus.cf".



Therefore, at this stage, we cannot say that our project is fully accessible to potential users, since we cannot fully transfer our files. However, as we aim, features such as viewing needs, registering, logging in, adding new needs for other people to see, etc. are available. So, after all files are transferred from local to internet area, it is fully possible for potential users to access our project via "needplusplus.cf" and use it as we planned.

10.2 Accessibility

On the other hand, ISO 9241-11, defines usability as: The “extent to which a product can be used by specified users to achieve specified goals effectively, efficiently and with satisfaction in a specified context of use”. This could address accessibility when: “specified users” includes people with a range of disabilities, and “specified context of use” includes accessibility considerations such as assistive Technologies. If we look at the subject from this perspective our application is quite accessible for the majority of disadvantaged people (disabled, not used to technology, etc.) and / or for disadvantaged situations (high daylight, dark, etc.). Our design colors are colors that will not affect the color blind, create discomfort and be understood; and these colors are suitable for viewing in high daylight and dark. The ease of use of our project is easy to understand for people who are not used to technology. Our understandable instructions and feedback for website forms and applications give good usability. At the current stage of our project, assistive technologies such as screen readers that read aloud content, and screen magnifiers that enlarge content or voice recognition software used to input text are not available. But these technologies are suitable for use.

11. REFERENCES

<https://www.quora.com/What-is-a-system-architecture-diagram-for-web-applications>

https://en.wikipedia.org/wiki/Systems_architecture

https://en.wikipedia.org/wiki/Software_design_pattern

<https://www.visual-paradigm.com/guide/uml-unified-modeling-language/how-to-model-mvc-with-uml-sequence-diagram/>

<https://en.wikipedia.org/wiki/Model%E2%80%93view%E2%80%93controller>

<http://siyainfo.com/2017/01/16/top-6-important-benefits-mvc-architecture-web-application-development-process/>

<https://www.codeupset.com/advantages-of-model-view-controller/>

<https://www.atlassian.com/continuous-delivery/principles>

<https://www.atlassian.com/continuous-delivery/principles/continuous-integration-vs-delivery-vs-deployment>

<https://www.w3.org/WAI/fundamentals/accessibility-usability-inclusion/>

12. SOME LINKS ABOUT NEED++ PROJECT

Project Link: www.needplusplus.cf

Github Project Link: <https://github.com/EsraNurArıcan/Need--Project/>

Jira Project Link: <https://needplusplus.atlassian.net/browse/NEED>