Implemented Use Case Descriptions

Use Case ID:	0		
Use Case Name:	Register		
Created By:	Group 4	Last Updated By:	Group 4
Date Created:	16.03.2020	Date Last Updated:	11.04.2020
Actor:	Applicant		
Description:	Applicant registers to the system.		
Preconditions:	No preconditions		
Postconditions:	Account is successfully created and the user is logged in.		
Priority:	High		
Frequency of Use:	Once per user		
Normal Course of Events:	1-) User fills in the required input fields provided by the system with their information. 2-) He/She clicks on the ‘Register’ button. 3-) System will ask the user to provide his/her phone number to be able to send a verification code. 4-) Then the user will be directed to a page in which there will be an input field for the verification code. 5-) System will give 2 minutes for the user to enter his/her verification code. 6-) If the provided verification code is correct then the user will be directed to his/her dashboard page.		
Alternative Courses:	No Alternative Courses		

Exceptions:	0.EX.1. Given mail address is invalid. (Response: System displays a message to indicate the error) 0.EX.2. Given mail address is already taken. (Response: System asks user to enter a new mail address) 0.EX.3 Given password is too short/long, includes special characters etc. (Response: System warns the user to re-enter a new password combination) 0.EX.4. Given phone number is invalid. (Response: System displays a message to indicate the error) 0.EX.5. The user has not entered the verification code correctly into the corresponding verification code input field provided by the system. (Response: System will display an error message and redirect the user back to the registration page)
Includes:	No includes
Special Requirements:	User licence agreement
Assumptions:	The mail server in which the user's email address is registered, should be available. Verification code sms successfully reaches user's phone
Notes and Issues:	Department and Grad School accounts created manually from firebase authentication, no need to register but it's also possible to change the role of registered account manually from firebase if there is any need.

Use Case ID:	1		
Use Case Name:	Login		
Created By:	Group 4	Last Updated By:	Group 4
Date Created:	16.03.2020	Date Last Updated:	11.04.2020
Actor:	Applicant, Grad School, Department		
Description:	User logs in to the system.		
Preconditions:	No preconditions		

Postconditions:	User is logged in.
Priority:	High
Frequency of Use:	20 times per user in a year
Normal Course of Events:	1-) User enters his/her email and password to login. 2-) User clicks on the sign in button. 3-) User is directed to an index page that has a dashboard which contains authorized functions assigned to his/her role.
Alternative Courses:	1.a-) User follows the password reset procedure by clicking forget my password link. 2.a-) After the password reset process, the user logs into the system with his/her new password. 1.b-) If the user has not signed up to the system, click the register button to start the registration process. 2.b-) After registration, the user logs into the system. 1.c-) User clicks on either ‘Sign in with Google’ or ‘Sign in with Facebook’ button.If the user hasn’t registered before, a phone verification will be needed additionally. 2.c-) After the user logs in, he/she is directed to an index page that has a dashboard which contains authorized functions assigned to his/her role.
Exceptions:	1.EX.1 User tried to log in with an email address that does not exist in the system or entered a password wrong. Response: System informs the user that he/she entered an incorrect email address or an invalid password for that email address.
Includes:	No includes
Special Requirements:	No special requirements
Assumptions:	No assumptions
Notes and Issues:	No notes or issues

Use Case ID:	2
Use Case Name:	Create an application

Created By:	Group 4	Last Updated By:	Group 4
Date Created:	17.03.2020	Date Last Updated:	11.04.2020

Actor:	Applicant
Description:	Applicant fills in an application form to apply for a program.
Preconditions:	User’s identity has been authenticated with an Applicant account.
Postconditions:	Applicant filled all the required parts of the application form and submitted the form successfully.
Priority:	High
Frequency of Use:	Once a year
Normal Course of Events:	1-) Applicant fills the given form by the system with correct and up-to-date information. 2-) Clicks on the ‘Next’ button. 3-) System directs the user to the next application page in which the user is asked to provide his/her related documents along with additional information. 4-) User selects his/her citizenship and working status. 5-) Uploads all required documents. (Use Case ID 3) 6-) Clicks on the submit application button.(Use Case ID 3) 7-) System saves the documents and additional information provided, and displays a message indicating the success of the transaction of the user. (Use Case ID 3)
Alternative Courses:	No alternative courses
Exceptions:	2.EX.1. User fills in a field with an invalid value or doesn’t fill at all. (Response: Ask the user to re-enter the field). 2.EX.2 User hasn’t uploaded all the required documents and tries to submit. (Response: System will decline the submission, and will inform the user to upload all the necessary documents).
Includes:	Use Case ID 3 (Upload required documents)
Special Requirements:	No special requirements.
Assumptions:	1-) Applicants have a minimum level of degree (If applying for a Master’s degree then the applicant must at least have a Bachelor’s degree) in order for him/her to be able to apply.

Notes and Issues:	No notes or issues

Use Case ID:	3		
Use Case Name:	Upload required document		
Created By:	Group 4	Last Updated By:	Group 4
Date Created:	16.03.2020	Date Last Updated:	13.04.2020

Actor:	Applicant
Description:	Applicant uploads the required document.
Preconditions:	User’s identity has been authenticated with an Applicant account. Applicant must have provided his/her personal information and clicked the ‘Next’ button in the first application page.
Postconditions:	Applicant has uploaded the required document successfully.
Priority:	High
Frequency of Use:	Once per document / per application

Normal Course of Events:	1-) Clicks on the “choose file” button on the upload document field. 2-) File explorer window for user’s device shows up. 3-) User chooses his/her documents from his/her device.. 4-) User realizes the upload is done when the file name appears instead of ‘Choose a File’ text. 5-) Clicks on the submit application button. 6-) System saves the documents and additional information provided, and displays a message indicating the success of the transaction of the user.
Alternative Courses:	No alternative courses
Exceptions:	3.EX.1 User uploads a document in the wrong format and tries to submit. (Response: System will decline the submission, inform the user to change the file). 3.EX.2 User uploads a document which exceeds the given upload capacity. (Response: System will warn the user with “Upload limit is exceeded.” message)
Includes:	No includes
Special Requirements:	No special requirements
Assumptions:	Applicants must have a minimum level of degree (If applying for a Master’s degree then the applicant must at least have a Bachelor’s degree) in order for him/her to be able to apply.
Notes and Issues:	No notes or issues

Use Case ID:	4		
Use Case Name:	Application Status		
Created By:	Group 4	Last Updated By:	Group 4
Date Created:	16.03.2020	Date Last Updated:	11.04.2020
Actor:	Applicant		

Description:	Applicant can check current progress of his/her application, Interview information and the results when they are announced.
Preconditions:	User’s identity has been authenticated with an Applicant account.
Postconditions:	Corresponding application status is displayed to the user.
Priority:	Low
Frequency of Use:	Once a day after an application is created
Normal Course of Events:	1-) User clicks on the ‘Application Status’ button, displayed on the left of the screen. 2-) System shows his/her application status. If the interview is set by the department it is shown right below the application status information. If the interview is not set yet, then an indicative message is displayed instead.
Alternative Courses:	No alternative courses
Exceptions:	4.EX.1 If the user has not created any application and clicks on the ‘Application Status’ (Response: A popover message will be displayed warning the user and he/she will not be directed to the application status page).
Includes:	No includes
Special Requirements:	No special requirements
Assumptions:	No assumptions
Notes and Issues:	No notes or issues

Architecture of the Implementation

Register

signUp() function gets the text fields and opens up a new page for the phone verification process by calling verify() method, if verification is successful then the page is closed and createAccount() function is called with text fields as its arguments.

verify() function calls the getCookie() method to get the phone number and sends a message that contains verification code to the given phone number.

createAccount(email,password,firstName,lastName,phone) function calls createUserWithEmailAndPassword(email,password) function from Firebase Authentication Service and if the authentication process is successful then Firebase Authentication Service sets the account as currentUser and calls the populateUser (currentUserId,firstName, lastName,phone).

-populateUser(userId,firstName,lastName,phone) creates a http request and opens a dashboard page for that specific user.

Login

signInWithEmail() function handles email login by reading the fields and calling firebase.auth().signInWithEmailAndPassword(email, password) which authenticates user if credentials are right.

signInWithGoogle() function handles Google login, uses Google login API. If its first login attempt, verifies account with phone verification pop-up.

signInWithFacebook() function handles Facebook login, uses Facebook login API. If its first login attempt, verifies account with phone verification pop-up.

getRegisterPage() function redirects user to register page

Create an Application

storeAndGetNextPage() function gets the text fields and assigns them to 'personalInfo' variable then it calls checkPersonalInfoFields(personalInfo) function and if that function returns true, the user will directed to the next application page.

checkPersonalInfoFields(personalInfo) function checks if the personalInfo fields and returns true if the fields are proper, false otherwise.

submit() function firstly calls checkDocumentFields() if it returns true, uses getCookie(personalInfo) method to get personal information then' it calls getInfo() function to get the documents information and creates an 'info' variable to store both 'personalInfo' and 'documentsInfo'. Then it calls updateDocument(uid, appId, file) function for each required document. Lastly, if all updateDocument functions are successful then it directs the user to the My Application Page.

uploadDocument(uid, appId, file) function uploads the given document to the firebase cloud storage by specifying user, application and the document.

checkDocumentFields(personelInfo) function checks if fields are filled correctly if not throws an error that notifies the user with the wrong field.

getInfo() function collects filled information from fields

getCookie(cname) function gets filled information from the page before.

Application Status

getApplicationData() function gets all the necessary information for logged in user's applications from the database.

addDataToTables(applicationData, departmentId) function fills view (html) tables with fetched information from the database.

Technologies

This demo is implemented using the technologies given below:

Languages: HTML, CSS, Javascript

CSS Frameworks: Bootstrap 4.4

Services: Firebase (Realtime Database, Authentication, Cloud Storage, Cloud Functions, Hosting)

APIs: Google Login API, Facebook Login API

Development and testing: Node.js, Firebase CLI Tools and Firebase Console.

Source Code Editor: Visual Studio

Software Development Version Control: Git (Github)

HTML (HyperText Markup Language) is the most basic building block of the Web. It defines the meaning and structure of web content. All the paragraphs, sections, images, headings and text are written in HTML. The content appears on the website in the order they are written in HTML.

CSS (Cascading Style Sheets) is a style sheet language used for describing the presentation of a document written in a markup language like HTML. It enables the separation of presentation and content, including layout, colors, and fonts.We used CSS because It takes the website from plain text elements to colorful designs.

Javascript is a scripting language that enables us to create dynamic activity on apps such as dynamically updating content, control multimedia and animate images. As a multi-paradigm language; it supports functional, event-driven and imperative programming styles. Scripting in JavaScript is what controls functions when buttons are clicked, how password forms are authenticated, how media is controlled e.g.

Bootstrap helps us quickly design and customize responsive sites, it's a front-end open source toolkit. It contains CSS and JavaScript-based design templates for typography, forms, buttons, navigation, and other interface components.

Node.js is a JavaScript runtime environment that executes JavaScript code outside of a web browser. We used Node.js because it lets us use JavaScript to write command line tools and for server-side scripting and running scripts server-side to produce dynamic web page content before the page is sent to the user's web browser.

Firebase CLI Tools adds all necessary Firebase service libraries to our project. Also it lets us deploy our project to the Firebase hosting server.

Firebase Console lets us manage all firebase services (Realtime Database, Authentication, Cloud Storage, Cloud Functions, Hosting) we use from one place.

Visual Studio Code is a source code editor with many features like debugging, syntax highlighting, intelligent code completion, snippets, code refactoring, and embedded Git.

Git is used as a distributed version-control system for tracking changes in source code during software development. It helps us coordinate work among us.

Github provides hosting for software development version control using Git.