**SOFTWARE REQUIREMENTS SPECIFICATION**

**for**

**LEARN&JOY**

**Prepared by**

**SEMA ALTINIŞIK**

**Version 1.0**

**20.05.2022**

**Table of Contents**

**1. Introduction………………………………………………………………………………….1**

1.1. Purpose

1.2. Scope

1.3. Definitions, Acronyms and Abbreviations

1.4. References

1.5. Overview

**2. General Description…………………………………………………………………………2**

2.1. Product Perspective

2.2. Product Functions

2.3. User Characteristics

2.4. General Constraints

2.5. Assumptions and Dependencies

**3. Specific Requirements………………………………………………………………………3**

3.1. External Interface Requirements

3.1.1. User Interfaces

3.1.2. Hardware Interfaces

3.1.3. Software Interfaces

3.1.4. Communication Interfaces

3.2. Functional Requirements

3.2.1. Features on Landing Page

3.2.1.1. Description and Priority

3.2.1.2. Stimulus/Response Sequences

3.2.1.3. Functional Requirements for Landing Page

3.2.2. Features on Content Page for User

3.2.2.1. Description and Priority

3.2.2.2. Stimulus/Response Sequences

3.2.2.3. Functional Requirements for Content Page for User

3.2.3. Features on Content Page for Admin

3.2.3.1. Description and Priority

3.2.3.2. Stimulus/Response Sequences

3.2.3.3. Functional Requirements for Content Page for Admin

**4. Other Non-Functional Requirements………………………………………………………4**

4.1. Performance Requirements

4.2. Security Requirements

4.3. Software Quality Attributes

**1. INTRODUCTION**

**1.1 Purpose**

In this document, the purpose and features of the 'Learn&Joy' application are mentioned. Learn&Joy is a online test application with test contents divided into categories. It can be used in many fields such as education, health etc.

**1.2 Scope**

The usage area of Learn&Joy is wide. It can be used in a social field such as education, health, or institutions, organizations.

**1.3 Definitions, Acronyms, and Abbreviations**

Learn&Joy

**1.4. References**

**1.5. Overview**

**2. GENERAL DESCRIPTION**

**2.1. Product Perspective**

Since Learn&Joy is an online test application, registration is required for internet access and keeping logs of users.

**2.2. Product Functions**

The functions of the application are included**.**

1) For admin, adding, editing, deleting tests.

2) For admin, viewing users' ratings.

3) For admin, editing, adding and removing categories.

4) For admin, having all authority.

5) For user, being able to login, logout or register.

6) For user, selecting tests, viewing questions.

7) For user, clicking multiple or single answers.

8) For user, viewing true or false answers and scores.

**2.3. Users and Characteristics**

It may vary according to the fields. Event owners can use the app if it's used in schools, by teachers and students, or if it's used by a social community. Anyone with technology, simple skills can use it.

**2.4. General Constraints**

**2.5. Assumptions and Dependencies**

**3. SPECIFIC REQUIREMENTS**

**3.1. External Interface Requirements**

**3.1.1. User Interfaces**

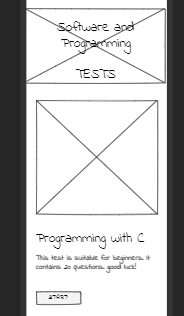
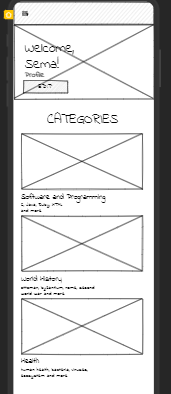
**3.1.1.1. Landing Page**

First, you place a login, registration system.

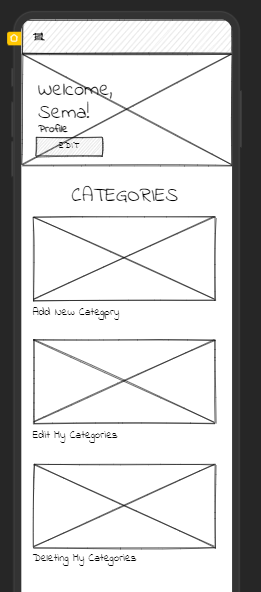
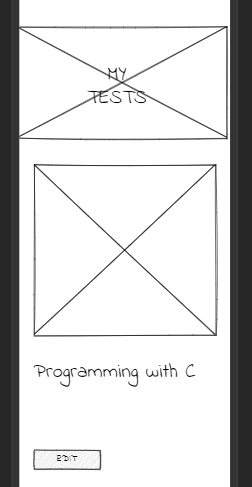
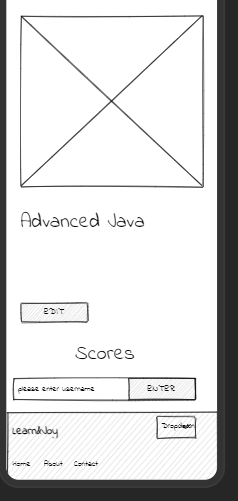


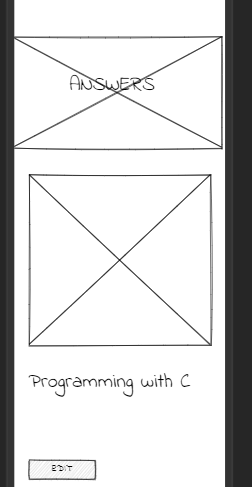
**3.1.1.2. Content Pages**

The content interface for users is as follows:



The content interface for admins is as follows:



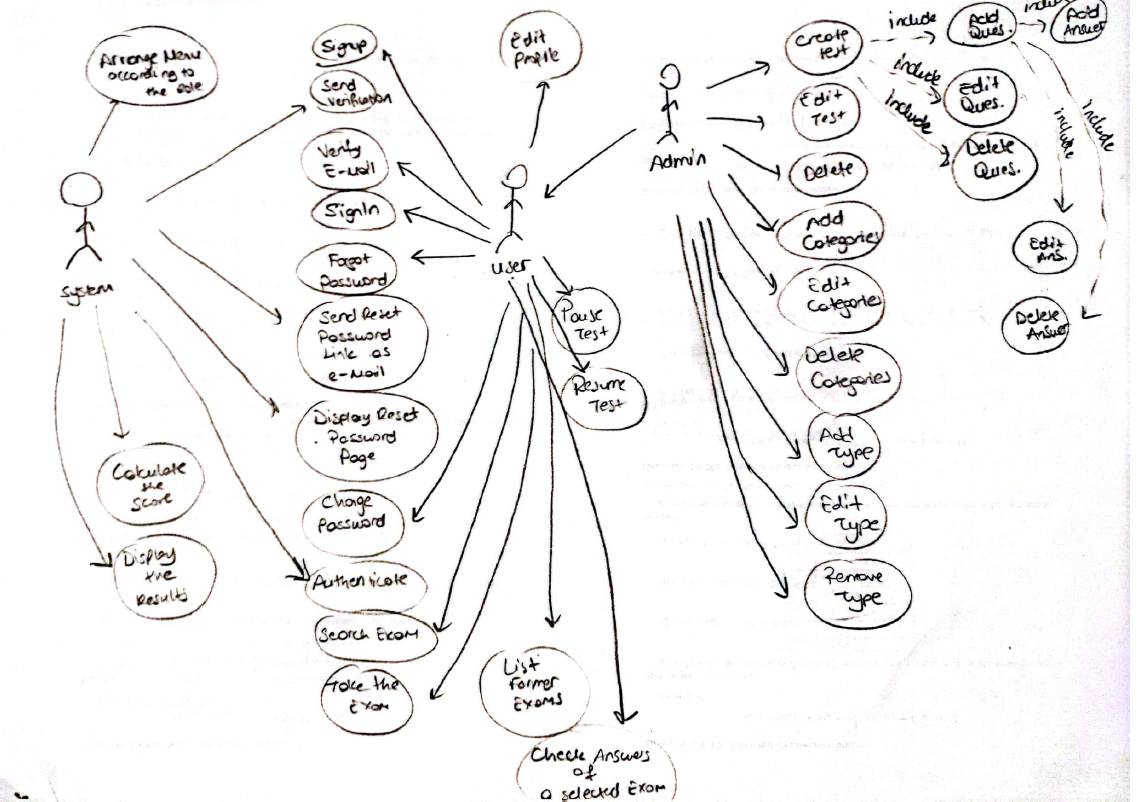


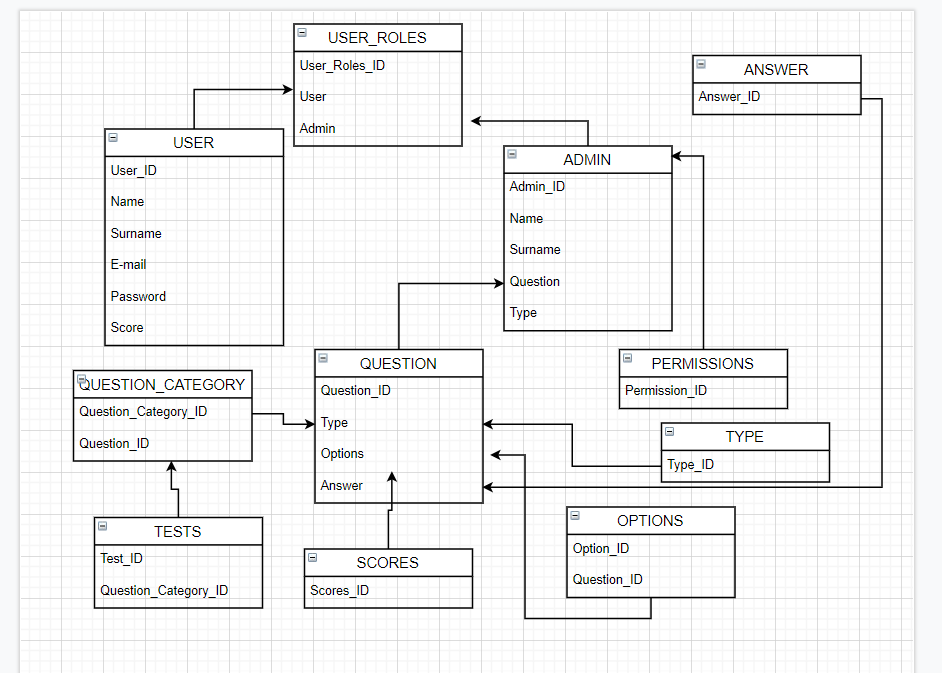
**3.1.2. Communications Interfaces**

If desired, a group can be established for the teams or the dm box feature can be added. (It was not added because it was not mentioned in the project summary.)

**3.2. Functional Requirements**

This section contains diagrams for function requirements.





**3.2.1. Login or Registering on Landing Page**

**3.2.1.1. Description and Priority**

The admin logs into the system. If the user is a member, s/he logs in, if not, he registers.

**3.2.1.2. Stimulus/Response Sequences**

Sequences of user actions and system responses is listed below:

1. User: Clicks the Login button.

2. System: Opens the Login tab.

3. User: Enters the username and password.

4. System: Checks password and nickname.

5. System: If the login is successful, the content page will open.

6. System: If false, it will print an error message to be re-entered.

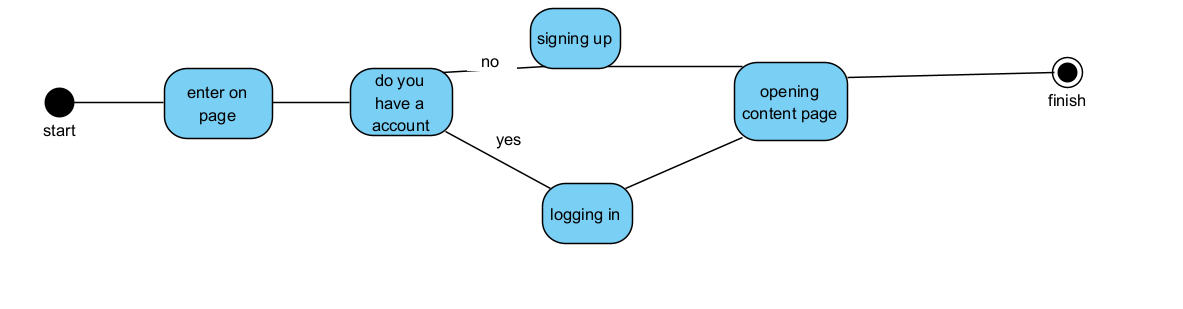
7. User: If he is not a member, he clicks the register button.

8. System: Opens the recording tab. asks the user for name, surname, age and e-mail.

9. User: Enters the requested information.

10. System: Saves information and creates profile.

11. System: Content screen opens successfully.

****

**3.2.1.3. Functional Requirements for Login or Registering on Landing Page**

REQ-1-) The user will open the landing page.

REQ-2-) The user will log into the application.

REQ-3-) After logging in, the content page will open.

REQ-4-) If he does not have an account, he will create a new account.

REQ-5-) After registering by e-mail, the login page will open.

**3.2.2. Content Page For User**

**3.2.2.1. Description and Priority**

The user can enter different categories, access the tests, solve the questions, see the

answers at the end of the test, view the score list.

**3.2.2.2. Stimulus/Response Sequences**

1. User: clicks on the category.

2. System: Shows the tests in the category.

3. User: chooses one of the tests.

4. System: opens the test and displays the questions.

5. User: Answers questions.

6. System: Records the user's responses.

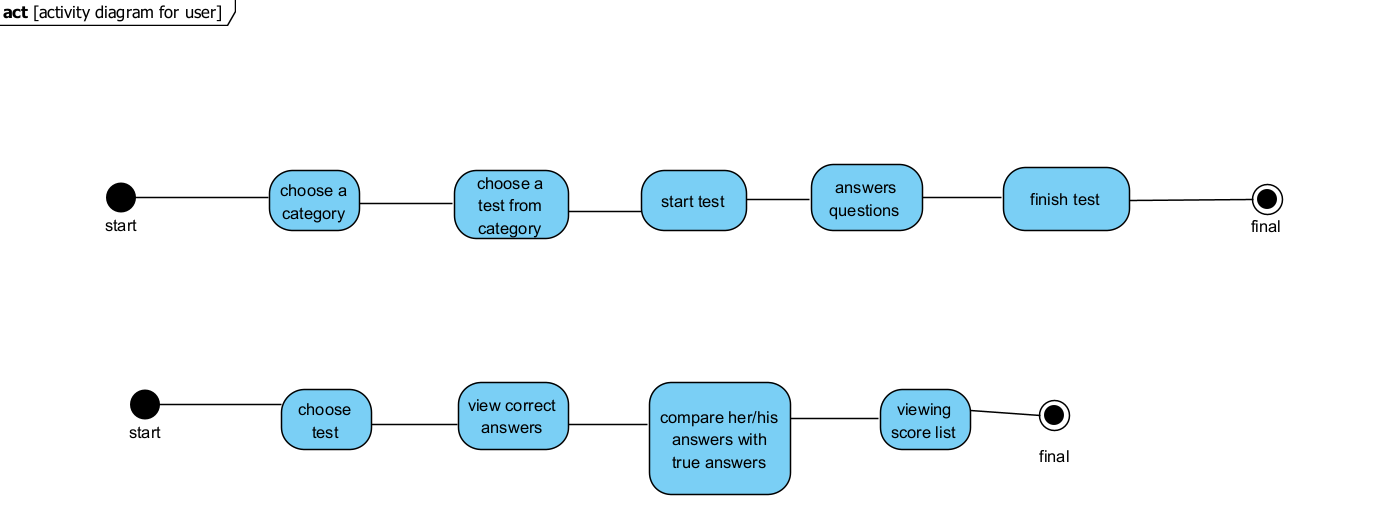
7. User: Finishes the test.

8. System: Checks the user's answers with the answer key.

9. System: Release the user's score.

10. User: Displays the score in the score list.

**3.2.2.3. Functional Requirements for User**



**3.2.2. Content Page for Admin**

**3.2.2.1. Description and Priority**

Admin edits, deletes or adds new categories. Adds, deletes, edits tests. You can see the answers to the questions. It can change. It can change the type, number, content of the question. Users can see the score list. He can access his points, ranges.

**3.2.2.2. Stimulus/Response Sequences**

1.

**4. OTHER NON-FUNCTIONAL REQUIREMENTS**

**4.1. Performance Requirements**

**4.2. Security Requirements**

**4.3. Software Quality Attributes**