Märchen

Software Requirements Specification

1.0

12.10.2020

Project Team

Kateryna Tkachenko
Dymusha Mykhailo
Tsoi Ihor
Yehor Kravchenko
Svitenko Heorhii
Shtets Kateryna

Revision History

Date	Description	Author	Comments
<date>	<Version 1>	<Your Name>	<First Revision>

Document Approval

The following Software Requirements Specification has been accepted and approved by the following:

Signature	Printed Name	Title	Date
	<Your Name>	Lead Software Eng.	
	A. David McKinnon	Instructor, CptS 322	

Table of Contents

1. Introduction

This Software Requirements Specification provides a complete description of all the functions and specifications of a learning platform “Marchen” for personal development of preschool children in the kindergartens. The expected audience of this document is the Product Owner, members of a team and other stakeholders.

1.1 Purpose

The purpose of this document is to build a developing platform for preschool children that will serve as a connecting bridge between children and adults (parents, tutors). Keeping its main focus on building social skills and helping children to adaptate to the world in an interesting and exciting manner, “Marchen” application also gives young people an opportunity to make their own decisions and apply received skills and knowledge in simulated scenarios. Evaluation of children’s answers helps to detect any personal deviations and problems that would be collected and reported to parents and tutors. By doing this, adults will be aware of the psychological state of their children and will be able to provide their children with special treatment.

1.2 Scope

The Marchen platform is meant to serve as a developing tool for children under 7. This system will be managed by parents and tutors in kindergartens mainly with the mobile application, but some functionality will be introduced in the Web version as well.

The main object of the Marchen platform is childrens’ personal growth. This platform will provide access to the library which contains choice-based fairy-tales. Children will be able not only to listen to the stories, but also change their plots by making decisions. Marchen facilitates interaction between system and child during the fairy tale by arising pop-ups with questions and possible answers. Moreover, the child's answers will be collected and evaluated to generate a personal portrait report after finishing the fairy tale. In addition, the platform

will be equipped with an automated award system in order to keep childrens' interest. Parents and tutors will be granted access to their childrens' reports and awards, as well as, fairy tales' library. Platform will allow parents to register and manipulate their childrens' profiles and tutors to create groups with existing children profiles.

To sum-up, Matchem platform is designed to provide an exciting learning process for children and help tutors and parents to see the inner personality of their children and treat their children in a more appropriate way.

1.3 Definitions, Acronyms, and Abbreviations

Term	**Definition**
Fairy tale	a children's story about magical and imaginary beings and lands
Plot	the main events of a play, novel, film, or similar work, devised and presented by the writer as an interrelated sequence
Pop up	a list of choices that is shown on a computer screen when the user requests it
Avatar	an icon or figure representing a particular person

1.4 References

R-1: IEEE Recommended Practice for Software Requirements Specifications Approved 25 June 1998

1.5 Overview

This document contains complete information about the project: its functions, interface, limitations.

The next chapter, the "General Description" section, is designed for the Product Owner of the project and gives an overview of the project business perspectives and functionality of the product. It describes the informal requirements such as user characteristics, general constraints and is used to establish a context for the technical requirements specification in the next chapter.

The third chapter, the "Specific Requirements" section, is written primarily for the developers and describes in technical terms the details of the functionality of the product.

Both sections of the document describe the same software product in its entirety, but are intended for different audiences and thus use different language.

2. General Description

This section will give an overview of the whole system. The system will be explained in its context to show how the system interacts with other systems and introduce the basic functionality of it. It will also describe what type of stakeholders that will use the system and what functionality is available for each type. At last, the constraints and assumptions for the system will be presented.

2.1 Product Perspective

“Marchen” was developed to help adults in the process of their children's growth. In contrast to other learning platforms, “Marchen” best serves for development of little children as it teaches them in an exciting manner and, what is more important, gives them an opportunity to build their individual personality by making decisions. It is an all-sufficient product and does not depend on other systems. However, third party libraries can be considered as an extension of the platform's functionality. Product will run on every browser and on mobile apps with Android platform. Also, users must be registered to have full access to the functionality.

2.2 Product Functions

The platform must provide users with registration and authorization processes. In the process of registration users define their roles: “Parent” or “Tutor”. After a successful authorization process the user's personal account is set and can be modified. Users with the role “Parent” can register their children and customize their profiles. Customization implies creation of an avatar and its personalization with the help of accessories. Users with the role “Tutor” can create group with existing children profiles. Both types of users can choose any available fairy tale and start it. Fairy tale is a story in audio format with pictures visualization and text. Questions connected with the plot of the story pop up as the fairy tale goes. Plot

evolves depending on the answers. Platform supports automated analysis of given answers. Each answer has its own weight that can be estimated in order to build a child's personal portrait. The total result is estimated after the ending of the fairy tale. Reports on each child's personal portrait are generated in a special section. Also, the platform is equipped with an award system that implies prizing a child with an award after finishing each fairy tale. Users must have full time access to all the functionality such as learning materials, reports and advice sections.

FE-1: Registration, authorization.

FE-2: Access to personal account page.

FE-3: Creation and manipulation of children's profiles.

FE-4: Creation of groups with existing children's profiles.

FE-5: Choice based fairy tales' library.

FE-6: Playing fairy tales and its manipulation.

FE-7: Answering pop-up questions and supporting plot changes.

FE-8: Achievement system.

FE-9: Answer analysis: formation of personal portrait of a child based on his answers during the fairy tale.

FE-10: Sending reports with recommendations to parents and tutors.

FE-11: Children profile customization (avatar personalization)

2.3 User Characteristics

Three kinds of users are envisaged by the system:

U-1: Unauthorized user: has access only to the registration page.

U-2: Authorized user with "Parent" role: have access to the library, reports and awards sections, additionally, can create and manipulate children's profiles.

U-3: Authorized user in "Tutor" mode: have access to the library, reports and awards sections, additionally, can create groups with existing children's profiles.

2.4 General Constraints

General constraints to the system:

C-1: Mobile app can be installed only on Android Platform.

C-2: Necessary access to Internet connection.

Constraints that are imposed on the technologies used for product Development:

C-3: Asp-Net Core for Back-End development

C-4: JavaScript, HTML5, CSS for Front-End development

C-5: SQL as a database choice

C-6: Kotlin for mobile development

2.5 Assumptions and Dependencies

AS-1: platform will be popular among parents of children under 7.

AS-2: platform will be popular among kindergartens.

D-1: Comfortable usage of the platform depends on browser or mobile and speed of Internet-connection.

D-2: Frequent usage of the platform will depend on child's interest and mood.

D-3: System only fits children whose age is under 7 years.

3. Specific Requirements

This section provides a detailed description of all inputs into and outputs from the system. It also gives a description of the hardware, software and communication interfaces and provides basic prototypes of the user interface.

3.1 External Interface Requirements

General requirements to external interfaces are:

R-1: The Android application shall support pop-up windows for notification.

R-2: The Application supports only english interface.

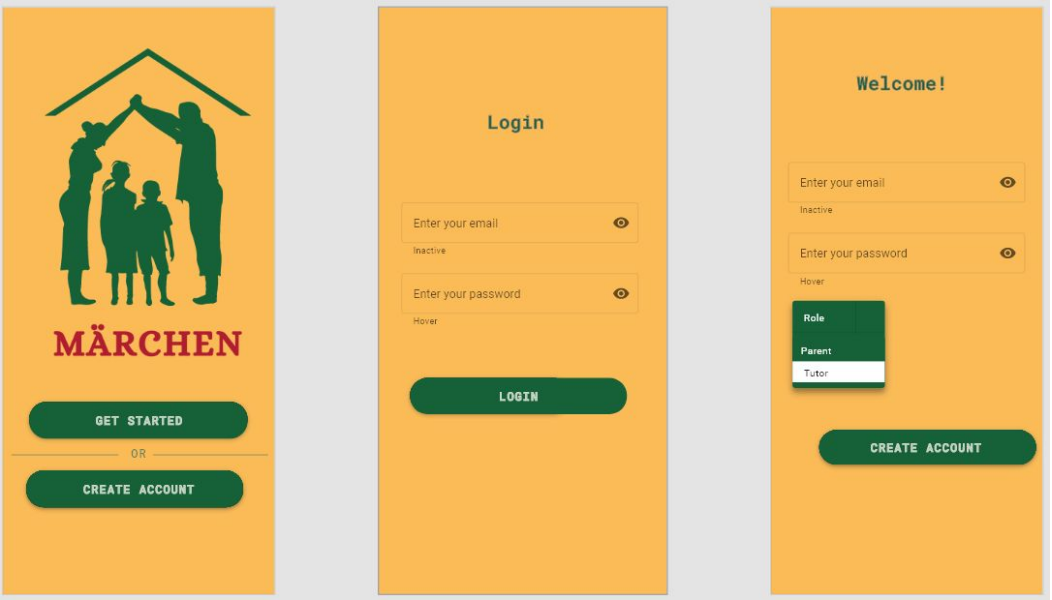
R-3: The user interface for the software shall be compatible with actual browsers such as Google Chrome, Opera, Safari, Firefox, Microsoft Edge.

R-4: The user interface shall be implemented using JavaScripts, HTML, CSS for web-application and Kotlin for mobile application.

R-5: The system shall support standard buttons, functions, or navigation links that will appear on every screen, such as policy, navigate bar, help description, tooltips.

3.1.1 User Interfaces

After entering the application user sees a page with possible entrance choices: login or create an account. For non registered users appears “Welcome page” and registered users goes to “Login Page” (img 3.1).

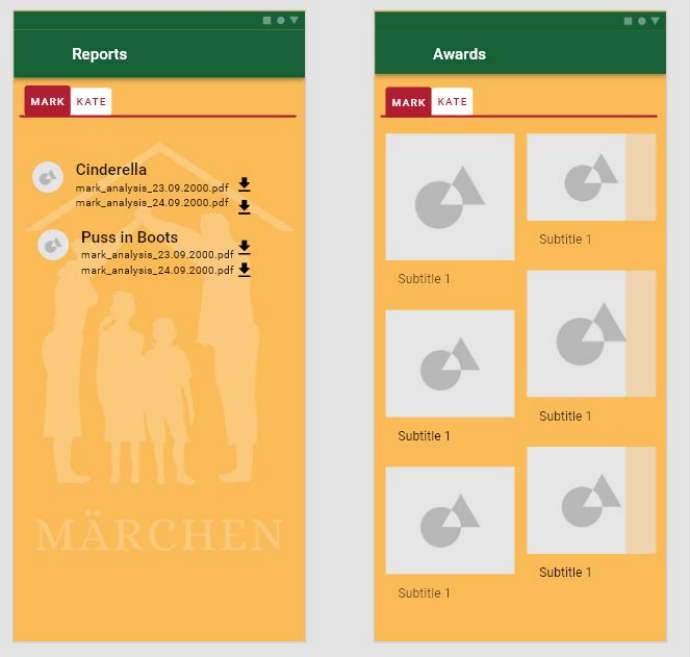


Img 3.1 - Registration/Login pages

After successful authentication, the user goes to the main page that contains all accessible functionality options (img 3.2).

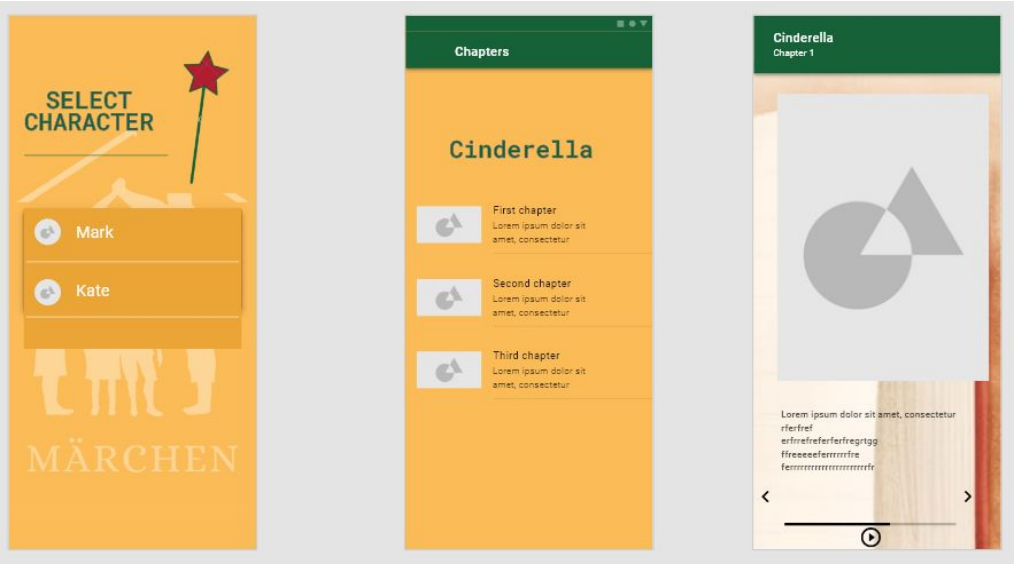
Img 3.2 - Main page

The Reports section contains all personal portrait analysis report files divided by children profile's. The Awards section contains all awards divided by children profiles. User, parent or tutor, sees only their children's profiles (img 3.3).



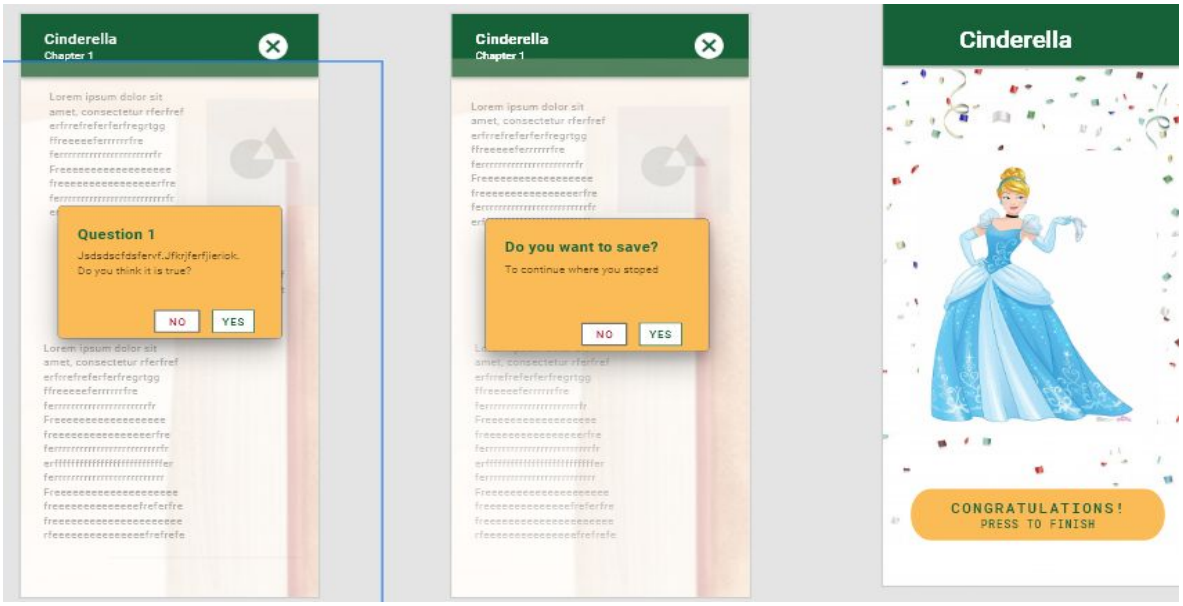
Img 3.3 - Reports and Awards pages

Before a fairy tale, the user, parent or tutor, chooses the child who will be listening to the story. Afterwards, the page with chapters of the fairy tale appears and it is followed by pages with the audio and visual components such as photo and text (img 3.4).



Img 3.4 - Fairy tale pages

In the process pop-ups with questions and answers appear. If a user wants to close the fairy tale, they will be asked if they want to save the state of the fairy tale. After finishing the fairy tale award picture appears on the screen (img 3.5).



Img 3.5 - Pop-ups and award page

3.1.2 Hardware Interfaces

To use an application you will need a mobile phone with Android system or device (PC, Laptop or smartphone) with Internet connection.

3.1.3 Software Interfaces

S-1: Android application shall communicate with server through API.
S-2: Web-application shall communicate with server through API.
S-3: Only one API for web-application and android application.

3.1.4 Communications Interfaces

C-1: The system shall use the HTTPs protocol for communication over the Internet.
C-2: The system shall use simple electronic forms for registering users and adding children's profiles.

3.2 Functional Requirements

This section describes specific features of the software project. If desired, some requirements may be specified in the use-case format and listed in the Use Cases Section.

3.2.1 <Registration>

3.2.1.1 Introduction

The user has the ability to register in the system.

3.2.1.2 Inputs

The user enters their email, password and chooses the role.

3.2.1.3 Processing

Input data is checked for validity, sent to the server, the server processes the data, sends a request to the database and checks if there is a user with the same email and confirms the registration.

3.2.1.4 Outputs

The user sees the confirmation of registration. If the user is successfully registered, they are sent to the login page.

3.2.1.5 Error Handling

Validators will process the data that the user has entered and, if there is an erroneous filling or missing required fields, will indicate this to the user.

System checks if there is a user with the same email, and if it is, the error message will be shown.

3.2.2 <Login>

3.2.2.1 Introduction

The user has the ability to login into the system.

3.2.2.2 Inputs

The user enters his email, password.

3.2.2.3 Processing

Input data is checked for validity, sent to the server, the server processes the data, sends a request to the database and checks if there is a user with the same login/email and confirms the logging in.

3.2.2.4 Outputs

If the user is successfully logged in, their profile's main page will be shown.

3.2.2.5 Error Handling

System checks if there is a user with the same login or email, and if it is not, the error message will be shown.

System checks if the password of a user is valid, and if it is not, the error message will be shown.

3.2.3 <Child's profile creation>

3.2.3.1 Introduction

The user with the role "Parent" has the ability to register their children in the system.

3.2.3.2 Inputs

The user enters the child's name, date of birth, and uploads the avatar picture.

3.2.3.3 Processing

Input data is checked for validity, sent to the server, the server processes the data, sends a request to the database and checks if there is a child with the same name and parent id and confirms the registration.

3.2.3.4 Outputs

If the child's profile is successfully registered, it will appear in the user account. .

3.2.3.5 Error Handling

System checks if a child with the same name and parent id exists, and if it is, the error message will be shown.

Validators will process the data that the user has entered and, if there is an erroneous filling or missing required fields, will indicate this to the user.

3.2.4 <Group creation>

3.2.4.1 Introduction

The user with the role "Tutor" has the ability to create a group by adding the existing children's profiles.

3.2.4.2 Inputs

The user enters the group name, finds the child by name and parent id and selects them.

3.2.4.3 Processing

Input data is checked for validity, sent to the server, the server processes the data, sends a request to the database and checks if there is a group with the same name and current user id and confirms the creation.

3.2.4.4 Outputs

If the group is successfully created, it will appear in the user account.

3.2.4.5 Error Handling

System checks if there is a group with the same name and current user id exists, and if it is, the error message will be shown.

System checks if a child with the same name and parent id exists, and if it is not, the error message will be shown.

Validators will process the data that the user has entered and, if there is an erroneous filling or missing required fields, will indicate this to the user.

3.2.5 <Report download>

3.2.5.1 Introduction

The user with the role “Tutor” or “Parent” has the ability to download the child’s personal portrait report.

3.2.5.2 Inputs

The user goes to page “Reports” and selects the report.

3.2.5.3 Processing

The server processes the data, PDF file is generated and downloaded.

3.2.5.4 Outputs

If the file is successfully generated, it will be downloaded to the device.

3.2.5.5 Error Handling

If any error will occur during the file generation, an error message will be shown to the user.

3.2.6 <Select the fairy tale>

3.2.6.1 Introduction

The user with the role “Tutor” or “Parent” has the ability to select the fairy tale.

3.2.6.2 Inputs

The user goes to the page “Library”, selects the fairy tale and selects the child who will listen to it.

3.2.6.3 Processing

The server processes the data, sends the request to the database and finds the fairy tale and child, fixes the start of the story session and redirects to the first page of the fairy tale.

3.2.6.4 Outputs

If the request is successfully handled, the user sees the first page of the fairy tale.

3.2.6.5 Error Handling

If any error will occur during the request, an error message will be shown to the user.

3.2.7 <Stop the play>

3.2.7.1 Introduction

The user with the role “Tutor” or “Parent” has the ability to stop the audio of the fairy tale.

3.2.7.2 Inputs

The user presses the icon “play” and audio stops playing.

3.2.7.3 Processing

The server processes the request and audio is stopped.

3.2.7.4 Outputs

If the request is successfully handled, the audio is stopped.

3.2.7.5 Error Handling

If any error will occur during the user trying to stop the audio, the user should be informed of the error.

3.2.8<Save the fairy tale >

3.2.6.1 Introduction

The user has the ability to close the playing fairy tale and save its state.

3.2.6.2 Inputs

The user presses the icon “Close” and the modal with confirmation question “Do you want to save the game?” user presses “Yes” button.

3.2.6.3 Processing

The server processes the request and saves the state of the fairy tale for the current child.

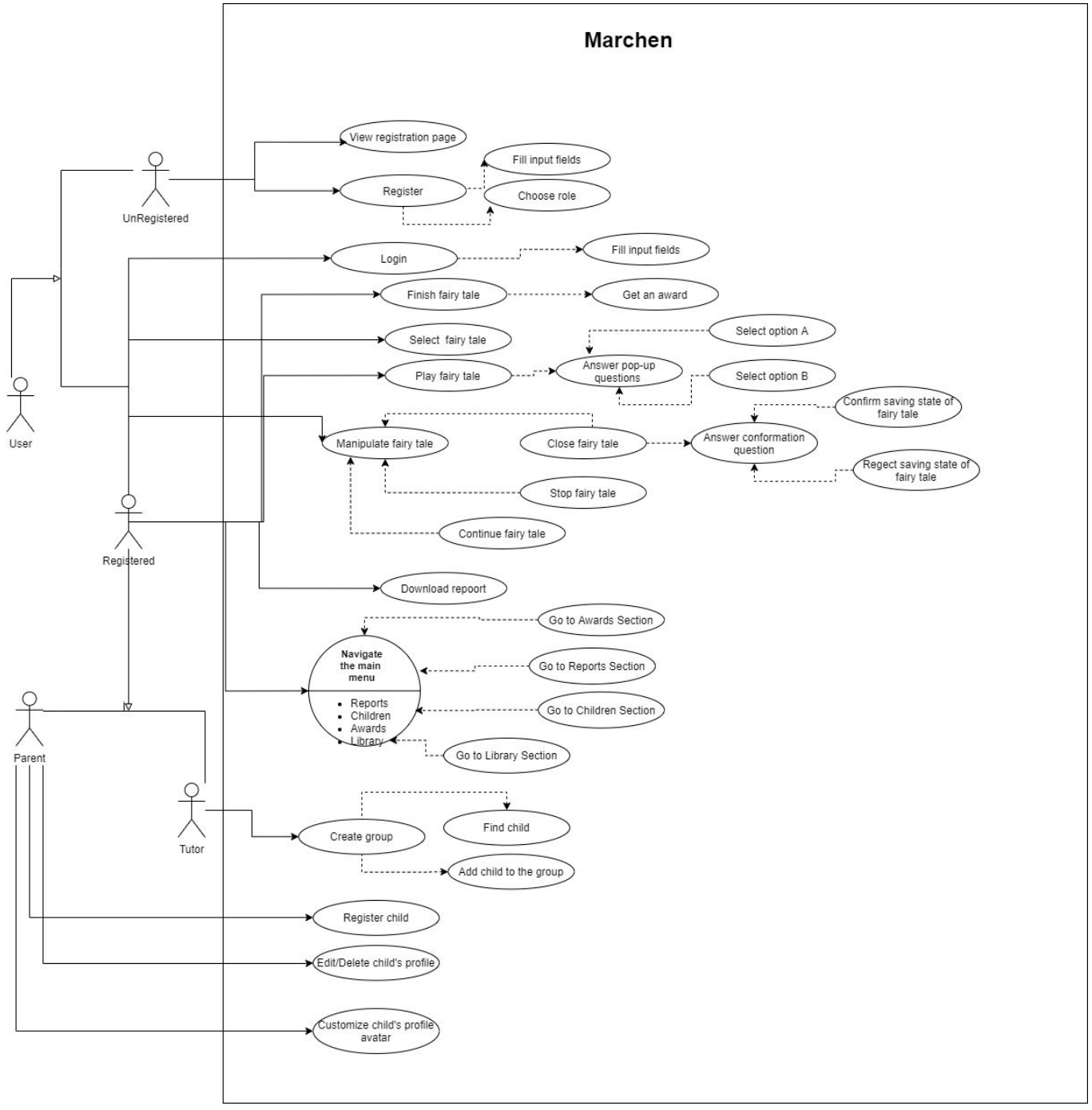
3.2.6.4 Outputs

If the request is successfully handled, the user is redirected to the “Library” page.

3.2.6.5 Error Handling

If any error will occur during the user trying to close the fairy tale , the user should be informed of the error.

3.3 Use Cases



3.4 Classes / Objects

3.4.1 <Class / Fairy tale>

3.4.1.1 Attributes

- Id:int
- Title:string
- Reward:string

- Slides IEnumerable:IEnumerable<Slides>

3.4.1.2 Functions

- Constructor

3.4.2 <Class / Object Slide>

3.4.2.1 Attributes

- Id: int
- FairyTaleId: int
- Image: string
- Text: string
- QuestionId: int

3.4.2.2 Functions

- Constructor

3.4.3 <Class / Object Question>

3.4.3.1 Attributes

- SlideId: int
- Text: String
- ChoiceOptionsId: int

3.4.3.2 Functions

- Constructor

3.4.4 <Class / Object ChoiceOptions>

3.4.4.1 Attributes

- Id: int
- Text: String
- MoveToSlideId: int

3.4.4.2 Functions

- Constructor

3.4.5 <Class / Object FairyTaleService>

3.4.4.1 Attributes

- _uow: IUnitOfWorkk
- _logger: ILogger

3.4.4.2 Functions

- GetAllFairyTales();
- SaveSate(int id);
- GetSlide(int id);
- PlayFairyTale(int id);

3.4.5 <Class / Object ReportService>

3.4.4.1 Attributes

- _fairyTaleService: IFairyTaleService
- _logger: ILogger

3.4.4.2 Functions

- GenerateReport();
- EstimateAnswers();

3.5 Non-Functional Requirements

3.5.1 Performance

1. The server shall take initial load time up to 2 minutes.
2. All Web pages generated by the system shall be fully downloadable in no more than 5 seconds over a 10Mbps modem connection.
3. Responses to queries shall take no longer than 2 seconds to load onto the screen after the user submits the query.
4. Database transactions shall take no longer than 1 second.

3.5.2 Reliability

1. The database shall use transactions and roll back all related updates when any update fails.
2. The database should be synchronized to cloud after each transaction.
3. Application will be available 99% of the time.

3.5.3 Availability

New module deployment will impact web pages of a system. The period of maintenance mustn't take longer than one hour. At this time users shall get a message about maintenance work.

3.5.4 Security

1. The system shall allow the user to perform actions via his personal account after authorization.
2. The system shall automatically log out all customers after a period of inactivity.
3. The system shall confirm all transactions with the customer's web browser or mobile device.
4. The system shall encrypt all user passwords.

3.5.5 Maintainability

1. System failure recovery shall be within an hour.
2. Finding bugs should happen within three minutes because of the logging system.

3.5.6 Portability

1. System shall be accessible on Android mobile devices with OS version 5.0 and higher.

2. System shall be accessible on Google Chrome, Opera, Safari, Firefox, Microsoft Edge.

3.6 Inverse Requirements

1. System shall not use red color for design interface in all situations besides critical errors or invalid user input.
2. Application shall not be compatible with Android devices v.4.4.4 and older.

3.7 Design Constraints

3.7.1 Software Languages

All back-end coding will be done in C# with using Asp.Net Core framework. Front-end will be done using Javascript language and HTML (with using Bootstrap). Android applications will be done using Kotlin.

3.7.2. Environments

During the development the next Integrated Development Environments will be used: Visual Studio, Visual Studio Code, Microsoft Server Management Studio, Android studio.

3.8 Logical Database Requirements

Will a database be used? If so, what logical requirements exist for data formats, storage capabilities, data retention, data integrity, etc.

3.9 Other Requirements

Catchall section for any additional requirements.

4. Analysis Models

List all analysis models used in developing specific requirements previously given in this SRS. Each model should include an introduction and a narrative description. Furthermore, each model should be traceable the SRS's requirements.

4.1 Sequence Diagrams

4.3 Data Flow Diagrams (DFD)

4.2 State-Transition Diagrams (STD)

5. Change Management Process

Identify and describe the process that will be used to update the SRS, as needed, when project scope or requirements change. Who can submit changes and by what means, and how will these changes be approved.

A. Appendices

Appendices may be used to provide additional (and hopefully helpful) information. If present, the SRS should explicitly state whether the information contained within an appendix is to be considered as a part of the SRS's overall set of requirements.

Example Appendices could include (initial) conceptual documents for the software project, marketing materials, minutes of meetings with the customer(s), etc.

A.1 Appendix 1

A.2 Appendix 2