

BAHRIA UNIVERSITY KARACHI CAMPUS

DEPARTMENT OF COMPUTER SCIENCE

(ARTIFICIAL INTELLIGENCE)



(SEN-220) Software Engineering

COUNTBUDDY

SOFTWARE REQUIREMENTS SPECIFICATION (SRS)

NAME	ENROLMENT
Syed Muhammad Yamann	02-136212-001
Muneeza Iftikhar	02-136212-012
Hafsa Hafeez Siddiqui	02-136212-026
Muhammad Anas	02-136212-033
Aqsa Khan	02-136212-039
Hiba Jamal	02-136212-049
Humaiz Javed	02-136212-051

COURSE INSTRUCTOR: ENGR. REEMA QAISER KHAN

Contents

1. Introduction	3
1.1 Purpose.....	3
1.2 Scope.....	3
1.3 Intended Audience and Intended Use	3
1.4 Document Conventions	3
1.4.1 Terminology.....	3
1.4.2 Units and Measurements	3
1.5 References	4
2. Overall Description.....	4
2.1 Product Perspective	4
2.2 Product Features.....	4
2.2.1 Functional Features.....	4
2.3 User Classes and Characteristics.....	4
2.3.1 Primary Users: Children with Dyscalculia	4
2.3.2 Secondary Users	5
2.4 Operating Environment.....	5
2.5 Design and Implementation Constraints	5
2.6 Assumptions and Dependencies	5
3. Project Context	6
3.1 Milestones	6
3.2 Development Strategy	6
3.3 Risk Analysis	6
3.4 Tools and Technologies	6
4. System Features and Requirements	7
4.1 Functional Requirements.....	7
4.2 Non-Functional Requirements	7
5. Test & Use Cases.....	7
5.1 Use Case 1: Registration	7
5.2 Test Cases for Registration.....	8
5.2 Use Case 2: Learning Courses	8
5.3 Test Case: Learning Courses.....	9
5.4 Use Case 3: Generate Report	9
5.5 Test Cases for Generate Report.....	10
APPENDIX A: USE CASE MODELLING	11

1. Introduction

1.1 Purpose

The purpose of the **COUNTBUDDY** web application is to provide an interactive and engaging learning platform for children with Dyscalculia. The application focuses on improving mathematical skills, including Arithmetic, Geometry, and Number Series, through flashcard-based learning, evaluations, progress tracking, and AI-generated personalized insights for students to track their progress and then go back to a specific course and re-practice to improve their performance.

1.2 Scope

The **COUNTBUDDY** application is designed to support children with Dyscalculia by enhancing their mathematical skills through engaging and interactive methods. It empowers parents and educators with AI-driven insights into the child's progress while fostering a structured learning environment with courses, worksheets, and tests. The application ensures a seamless and enjoyable user experience for both children and their guardians.

1.3 Intended Audience and Intended Use

The target audience for the application includes children aged 5-7 with Dyscalculia who require support in learning fundamental mathematical concepts. It also caters to parents and guardians interested in monitoring their child's learning progress. Additionally, the application is designed for educators and therapists specializing in special needs learning, providing them with tools to better support their students.

COUNTBUDDY is an innovative educational tool designed to make learning engaging and effective for children. It offers flashcards to simplify complex math concepts, worksheets for each course for each child's progress, and AI-generated evaluations that track improvements and provide personalized recommendations for further practice. This combination of interactive and adaptive features ensures a structured yet enjoyable learning experience.

1.4 Document Conventions

This section outlines the conventions, standards, and notations used throughout the SRS document to ensure consistency and clarity.

1.4.1 Terminology

- **Dyscalculia:** A learning disorder that affects mathematical abilities and number comprehension.
- **Evaluation:** A structured test designed to assess a child's understanding of mathematical concepts.
- **Flashcards:** Interactive learning tools used in the app to teach concepts like arithmetic, geometry, and number sequences.
- **AI Report Generation:** After evaluation is completed, a report is generated for each child to view their performance.

1.4.2 Units and Measurements

- **Time:** Expressed in seconds (e.g., Timer: 2s for evaluation tasks).

- **Scores:** Presented as "Obtained Marks / Total Marks" (e.g., 4/15).

1.5 References

The following section contains the mood board ideas and UI design links used and referenced for COUNTBUDDY.

- **Prototype:** [COUNTBUDDY - final prototype](#)
- **Moodboard:** [Figma Link](#)

2. Overall Description

2.1 Product Perspective

The product is a web-based educational platform designed specifically for children with Dyscalculia, aiming to provide an interactive and supportive learning environment. The application will incorporate gamified learning modules and AI-driven analytics to enhance the user's mathematical abilities.

This system is an independent, stand-alone application with seamless integration options for parents, teachers, and healthcare professionals. The system leverages modern web technologies such as HTML-CSS and JavaScript for frontend, Flask for back-end, and AI modules to generate performance insights on areas of strength and improvement. The product is part of a broader initiative to support children with special learning needs by addressing the unique challenges posed by Dyscalculia. It integrates multimedia tools, interactive exercises, and visual learning aids to cater to diverse learning preferences.

2.2 Product Features

The following are the features present in the web application:

2.2.1 Functional Features

1. **Secure User Authentication:** Robust login systems to ensure data security and privacy.
2. **Interactive Learning Flashcard Modules:** Math lessons are structured with the concept of flashcard learning for children with Dyscalculia, focusing on key areas such as **arithmetic**, **geometry**, and **number series**.
3. **Evaluation (Tests):** The evaluation section is created to help children evaluate their math skills after their flashcard-based learning courses are completed. This evaluation helps with memory and to fully grasp the concepts behind every course.
4. **AI-Driven Insights and Reports:** The system tracks a child's performance and generates detailed insights into their strengths and weaknesses. It also provides learning recommendations to guide children towards improvement.

2.3 User Classes and Characteristics

2.3.1 Primary Users: Children with Dyscalculia

The primary users of the app are children aged 5–7 who face challenges in understanding numbers and mathematical concepts due to dyscalculia. They require an interactive, visually engaging, and user-friendly interface that simplifies learning. The app must adapt to their learning pace and

provide positive reinforcement to maintain their motivation and focus.

2.3.2 Secondary Users

1. Parents

Parents play a crucial role in monitoring their child's learning progress. They need access to an intuitive dashboard that provides detailed insights into their child's performance. Recommendations for additional activities or areas of improvement can further help them support their child's learning journey.

2. Educators/Tutors

Educators or tutors, who assist children with dyscalculia, require tools to monitor individual or group performance. They may also need the ability to assign specific exercises or track overall learning progress. The app should provide detailed analytics and reports to help educators make informed decisions.

3. Administrative Users: System Administrators

System administrators manage the app's backend operations. They are responsible for ensuring smooth functionality, resolving technical issues, and updating content. The app should provide secure access and efficient tools for managing users and maintaining system stability.

2.4 Operating Environment

The web app is designed to function on laptops and desktops with operating systems including Windows and macOS. It performs optimally on modern browsers such as Microsoft Edge, Google Chrome, and Mozilla Firefox. The backend is built using Flask and MySQL, ensuring efficient data handling and server-side functionality. While the app requires an internet connection, it is lightweight and does not demand high bandwidth, making it accessible even with moderate-speed internet connections.

2.5 Design and Implementation Constraints

The web app was developed using core web technologies, including HTML, CSS, JavaScript, Flask for the backend, and MySQL for the database. Due to limited resources, as it was developed by a team of students, and a strict time constraint from 9th October to 25th December, the app focuses on essential features and functionality within these limitations.

2.6 Assumptions and Dependencies

Assumptions

- Users will access the web app on laptops or desktops with modern operating systems (Windows or macOS).
- Users will have a stable internet connection, even if moderate in speed.
- The app will be used on supported browsers such as Microsoft Edge, Google Chrome, or Mozilla Firefox.

Dependencies

- The app relies on Flask for server-side development and MySQL for database management.
- Functionality depends on modern browsers for compatibility and rendering.
- The hosting environment must support Flask and MySQL integration for seamless operation.

3. Project Context

This section includes milestones, development strategy and tools.

3.1 Milestones

MILESTONE	START	FINISH
Documentation & Research Phase I	26, Sep 2024	8, Oct 2024
Course Outline & Flashcard Creation	12, Oct 2024	17, Oct 2024
UI Design	18, Oct 2024	20, Nov 2024
Front-End Development	21, Nov 2024	2, Dec 2024
Model Development	24, Nov 2024	9, Dec 2024
Back-End Integration	24, Nov 2024	11, Dec 2024
Testing & Quality Assurance	12, Dec 2024	19, Dec 2024
Documentation & Compilation Phase II	21, Dec 2024	25, Dec 2024
Project Showcase	26, Dec 2024	26, Dec 2024

3.2 Development Strategy

The development of the web app followed an incremental approach, where features were developed and tested in phases to ensure functionality and user satisfaction. The process began with defining core requirements and creating prototypes for key modules. Once the prototypes were validated, the team moved to full-scale development, focusing on essential features such as user interaction, data storage, and accessibility. Regular testing was conducted after each phase to identify and fix bugs promptly. The timeline was carefully divided into planning, development, and testing stages to meet the strict deadline of 26th December.

3.3 Risk Analysis

The development process faced several risks, including:

- **Technical Limitations:** As students, the team had limited experience with Flask and MySQL, which posed challenges during backend implementation.
- **Resource Constraints:** The project was developed with limited hardware and software resources, potentially impacting the app's scalability.
- **Time Constraints:** The tight deadline increased the likelihood of overlooking minor bugs or optimization opportunities.
- **Browser Compatibility:** Ensuring the app functions seamlessly across supported browsers required thorough testing to avoid potential compatibility issues. Mitigation strategies included regular team meetings, prioritizing essential features, and extensive browser testing to minimize these risks.

3.4 Tools and Technologies

The development process utilized the following tools and technologies:

- **Frontend:** HTML, CSS, and JavaScript for building an interactive and user-friendly interface.
- **Backend:** Flask for server-side development to handle requests and manage data.

- **Database:** MySQL for efficient data storage and retrieval.
- **Testing:** Browser-based debugging tools and manual testing to ensure functionality and compatibility.

4. System Features and Requirements

4.1 Functional Requirements

- **FR1:** The system shall allow children to access flashcard-based learning modules for Arithmetic, Geometry, and Number Series.
- **FR2:** The system shall track the user's course progress.
- **FR3:** The system shall unlock evaluation tests upon course completion.
- **FR4:** The system shall generate AI-based performance reports and recommendations.
- **FR5:** The system shall provide worksheets tailored to evaluation outcomes.

4.2 Non-Functional Requirements

- **Performance:** The application shall respond within 2 seconds for all user actions.
- **Scalability:** The system shall support up to 1000 concurrent users.
- **Security:** User data shall be encrypted, and access shall be secured via authentication.
- **Usability:** The interface shall be user-friendly and designed for children aged 5-7.
- **Availability:** The application shall be available 90% of the time.

5. Test & Use Cases

This section contains all the test and use cases of the web app.

5.1 Use Case 1: Registration

Use Case ID: UC-REG-01

Title: User Registration

Actors: New User

Description: This use case describes how a user registers for the application by providing necessary details.

Preconditions:

- The user has access to the registration page.
- The user is not already registered.

Steps:

1. User navigates to the registration page.
2. User enters the required details: Name, Age, Email, Password, and Confirm Password.
3. User clicks the "Sign Up" button.
4. The system validates the input data.
5. Upon successful validation, the user account is created.
6. The system displays a success message.

Postconditions:

- A new user account is created and stored in the database.
- The user is redirected to the login page or the dashboard.

5.2 Test Cases for Registration

Test Case ID	Feature	Test Case Title	Steps	Expected Result
TC-REG-01	Registration Functionality	Validate successful registration	1. Navigate to the registration page. 2. Enter valid details for all fields. 3. Click the "Sign Up" button.	A new user account is created, and a success message is displayed.
TC-REG-02	Registration Functionality	Validate registration with missing fields	1. Leave one or more fields empty. 2. Click the "Sign Up" button.	An error message prompts the user to fill in the missing fields.
TC-REG-03	Registration Functionality	Validate password mismatch during registration	1. Enter valid details but use different passwords in "Password" and "Confirm Password" fields. 2. Click the "Sign Up" button.	An error message indicating password mismatch is displayed.

5.2 Use Case 2: Learning Courses

Use Case ID: UC-COURSE-01

Title: Access Learning Courses

Actors: Registered User (Child)

Description: This use case describes how a user accesses and interacts with learning courses (Arithmetic, Geometry, Number Series).

Preconditions:

- The user is logged into the system.

Steps:

1. User logs into the application.
2. User navigates to the learning courses section.

3. User selects a course (e.g., Arithmetic).
4. The course content (flashcards, visuals, etc.) is loaded.
5. User interacts with the course elements (e.g., solving flashcards or exploring visuals).

Postconditions:

- The course data is marked as "completed" upon interaction.
- The user's progress is updated in the database.

5.3 Test Case: Learning Courses

Test Case ID	Feature	Test Case Title	Steps	Expected Result
TC-COURSE-01	Learning Courses	Validate course accessibility	1. Log in to the user account. 2. Click on any course (Arithmetic, Geometry, or Number Series). 3. Click "Start."	The course content is loaded and displayed correctly.
TC-COURSE-02	Learning Courses	Validate interactive elements in courses	1. Start a course (e.g., Arithmetic). 2. Interact with flashcards or other elements.	The interactive elements (flashcards, visuals) work correctly and provide feedback.

5.4 Use Case 3: Generate Report

Use Case ID: UC-REPORT-01

Title: Generate and Download Performance Report

Actors: Registered User (Parent/Child)

Description: This use case describes how a user generates and downloads their performance report after completing an evaluation.

Preconditions:

- The user has completed the evaluation.

Steps:

1. User logs into the application.
2. User completes the evaluation.
3. User clicks the "Generate Report" button.

4. The system processes the user's results and generates a report.
5. The "Download Report" button is enabled.
6. User clicks the "Download Report" button to save the report locally as a PDF.

Postconditions:

- The performance report is saved locally on the user's device.
- The system logs the report generation activity.

5.5 Test Cases for Generate Report

Test Case ID	Feature	Test Case Title	Steps	Expected Result
TC-REPORT-01	Generate Report	Validate report generation	1. Log in to the application. 2. Complete the evaluation. 3. Click "Generate Report."	The system generates a performance report with accurate data (scores, recommendations).
TC-REPORT-02	Generate Report	Validate report download functionality	1. After generating the report, click the "Download Report" button.	The system allows the report to be downloaded as a PDF file, which contains all metrics and recommendations.

APPENDIX A: USE CASE MODELLING

