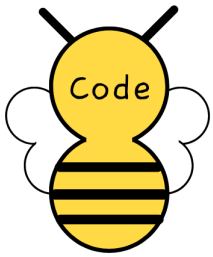
System Requirement
CodeBee

Hongqing Cao, 400053625
Jason Nagy, 400055130
Shivaansh Prasann, 400100351
Sida Wang, 400072157
Weidong Yang, 400065354
Yachen Wu, 400131465

November 1, 2020



Contents

List of Tables

List of Figures

Date	Version	Notes
Oct 29, 2020	0.1	Doc created with structure
Oct 29, 2020	1.0	Initial Draft
Oct 30, 2020	1.1	First Completed Draft
Oct 31, 2020	1.2	Added Definitions, Acronyms, and Abbreviations
Nov 1, 2020	1.3	Added Logo

Table 1: **Revision History**

1 Introduction

1.1 Purpose

The purpose of CodeBee is to a deliver a beginner friendly, gamified educational experience to students aged 10-16, in order to help them learn the basic concepts and thinking patterns necessary for computer programming. This document contains the software requirements and specifications for CodeBee. This document comprises of a system description, the system boundaries, an overview of the system behaviour, a functional decomposition of the system and details system requirements supported by appropriate rationale.

1.2 Scope

As an educational web application, the project will specifically focus on the enjoyability of programming-related computer games. The functionalities of the system that are included in the scope:

- Log-in system which allows users to Log-in/Sign-up with game progress stored within the account.
- A level management system to allow users to select and attempt different game levels of varying difficulty.
- A game system that provides meaningful and enjoyable programming challenges to the users in the form of an interactive game.
- An achievement system which serves as a reward mechanic that will be accessible to the user after a certain portion of the game has been completed.

Additionally, the following points are deemed to be out of scope:

- Allowing users to design their own levels for CodeBee.
- Using an in-house programming language and compiler within the application.

1.3 Definitions, Acronyms, and Abbreviations

Term	Definition
CodeBee	The name of the application
System, Application, Product	Different terms referring CodeBee
Sandbox	A game mode that allows the user to create freely without a goal
Cloud Server	A virtual server (rather than a physical server) running in a cloud computing environment. It is built, hosted and delivered via a cloud computing platform via the internet, and can be accessed remotely.

Table 2: **Definitions, acronyms, and abbreviations**

2 System Description

2.1 Overview

The product has been initiated by the Capstone team and has been conceived and supervised by Dr.Alan Wassyng and Dr.Kevin Browne as the objective of the final year capstone project for students in the Software and Computing Department at McMaster University. In the following context, the system will be introduced as an an interactive yet playful web application which helps young students learn and implement the basic concepts and logic of programming.

2.2 Users and Characteristics

2.2.1 Users

- Young students aged 10-16
- Programming educational institutions

2.2.2 Clients

- CodeBee Project Proposers
- Dr. Alan Wassyng, Dr. Kevin Browne, and Teaching Assistants

2.2.3 Users of the product

The product will be used as a web application to serve the purpose of educating youth about entry level programming, and the users can be potentially divided into two categories. The first category of users will be the educational institutions staff, who will manage and maintain the application. The other one will be teenagers who are willing to learn programming in a meaningful way.

2.3 Assumptions

- All users should have a device with a web browser and internet connection.
- All users have the basic knowledge of using a web browser.
- All users do not have visual impairments.

2.4 Controlled Variables

2.4.1 For a single user

- **username**: The username that the user registered.
- **password**: The password that the user registered.
- **unlockedLevels**: The game levels that are unlocked by the user.
- **lockedLevels**: The game levels that are not unlocked by the user.
- **unlockedAchievements**: The achievements that are unlocked by the user.
- **lockedAchievements**: The achievements that are not unlocked by the user.
- **selectedLevel**: The specific game level that is selected by the user at runtime.

2.4.2 For all users

- **numUsers**: The number of registered users.

2.5 Constants

- **numLevels**: The number of game level pre-defined in the system.
- **numAchievements**: The number of game achievements pre-defined in the system.

2.6 Normal Operation

As for normal operation of the system, it shall be able to allow the user to create and log in to their account, precisely select levels, then enjoy the educational games within the level selected. At the end, the achievement panel will be accessible once if a certain level of games is accomplished. As an additional feature, the application shall also be able to allow the user to go into the sandbox mode and explore it. See Use Case Diagram for more details.

2.7 Undesired Event Handling

- Connection lost due to unexpected internet issue
- Game process can not be stored due to Back-end Server maintenance
- Application crushes due to unfounded bugs

3 Diagrams

3.1 System Diagram

3.1.1 Abstract Level

This diagram shows the input and output of the CodeBee system in general. The dashed arrows represents input and output of the system.

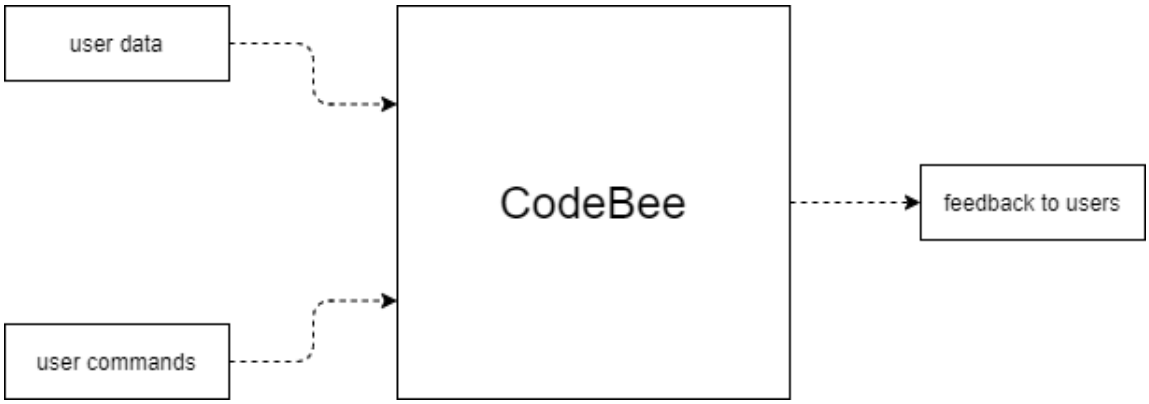


Figure 1: **Abstract System Diagram**

3.1.2 Detailed Level

This diagram shows the input and output of the subsystems in CodeBee specifically. The subsystems shown in the diagram are mapped to the subsystems specified in the **Functional Requirements Section**.The solid arrows represent input and output of the system. The dashed arrows in this diagram indicate invocation relationships between subsystems.

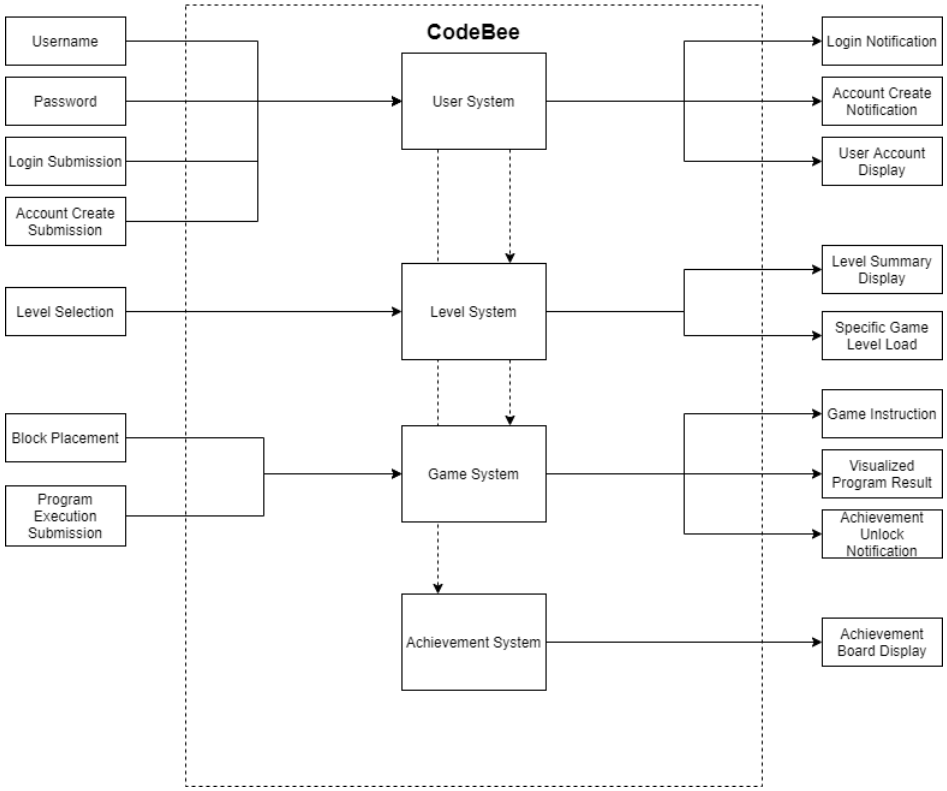


Figure 2: **Detailed System Diagram**

Here are some notes about entities in the diagram:

- **Username:** The **username** variable from the user input, which represents the account username.
- **Password:** The **password** variable from the user input, which represents the account password.
- **Login Submission:** The user fills in username and password, then submits the login form.
- **Account Create Submission:** The user fills in username and password, then submits the account creation form.
- **Level Selection:** The **selectedLevel** variable from the user input, which represents the selection of a specific level.
- **Block Placement:** The user drags a code block and places it in the coding area.
- **Program Execution Submission:** The user submits the code and activates the execution of it.
- **Login Notification:** The notification to inform the user that the login is successful/failed.
- **Account Create Notification:** The notification to inform the user that the account creation is successful/failed.
- **User Account Display:** The graphical representation of the user account.
- **Level Summary Display:** The graphical representation of the all game levels (including **unlockedLevels** and **lockedLevels**) where the user can select a specific game level or play sandbox game mode.
- **Specific Game Level Load:** The graphical representation of a specific game level or the sandbox game mode.
- **Game Instruction:** The text on the graphical representation of the game, one which contain any information required for the user to successfully complete the level.
- **Visualized Program result:** The visualized program result(some graphics) on the graphical representation of the game, one which shows that the level is successfully completed/failed.
- **Achievement Unlock Notification:** Upon successfully completion of one/many levels, a notification will be displayed to inform the user an achievement is unlocked.
- **Achievement Board Display:** The graphical representation of all unlocked achievements represented by **unlockedAchievements**.

3.2 Use Case Diagram showing Behaviour Overview

This diagram represents the user behaviour with each main functions in CodeBee with the following definitions:

- The rectangle is system boundary.
- Each ellipse is a use case.
- Solid ellipses can be performed by users, dotted ellipses are unreachable.
- Solid lines represent the use case that is directly invoked by the user.
- Dotted lines with the flag "<extend>" represent an operation can be performed by the user from the previous use case.
- Dotted lines with the flag "<include>" are automatically operated by the system.
- Dotted lines with other flags stand for an operation with specified condition.

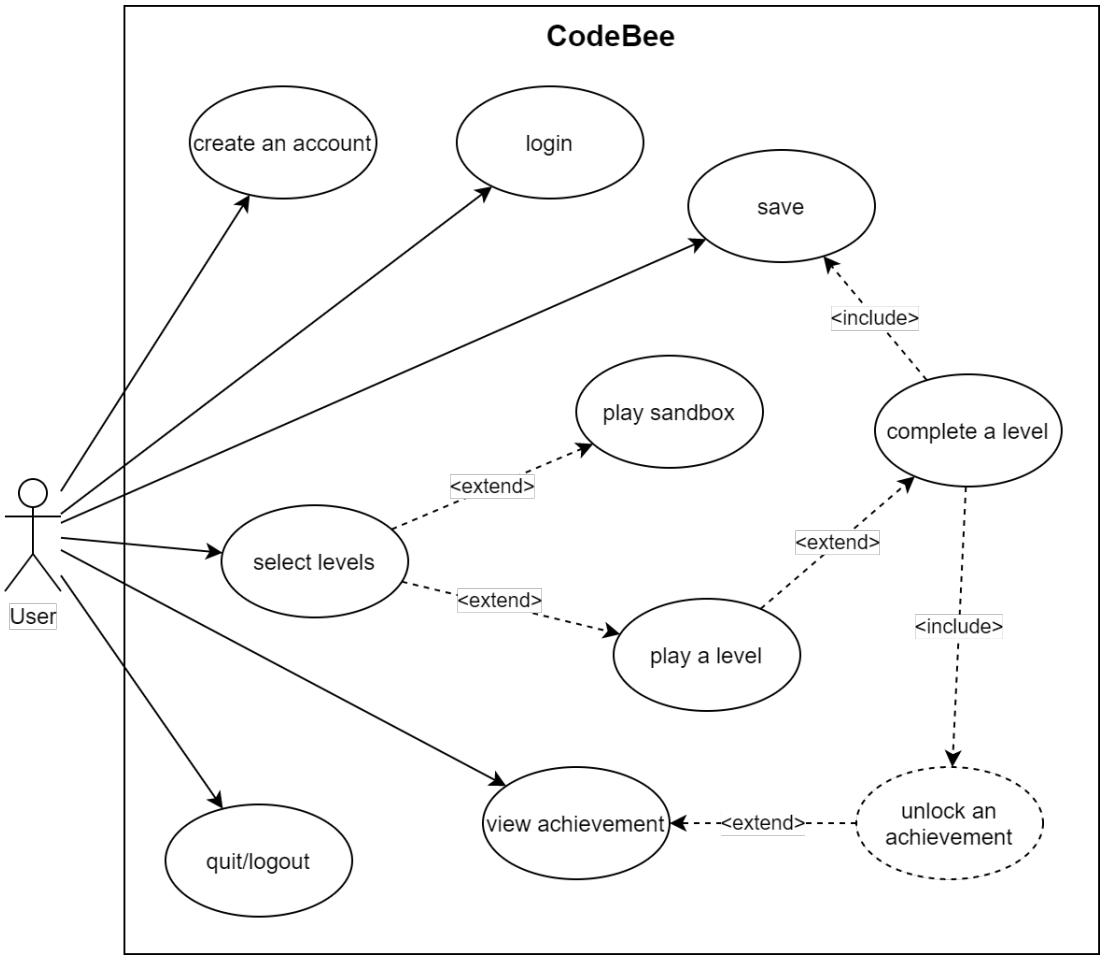


Figure 3: **Use Case Diagram**

Each uses case is corresponding to one or several functional requirements.

- **create an account:** FR1.1
- **login:** FR1.2
- **play sandbox:** FR2.6
- **save:** FR2.7, FR3.20
- **select levels:** FR2.2
- **play a level:** FR3.1∼FR3.17, FR3.19
- **complete a level:** FR3.18
- **view achievement:** FR4.3
- **unlock an achievement:** FR4.1∼FR4.2, FR4.4
- **quit/logout:** FR1.9

3.3 Context Diagram showing System Boundaries

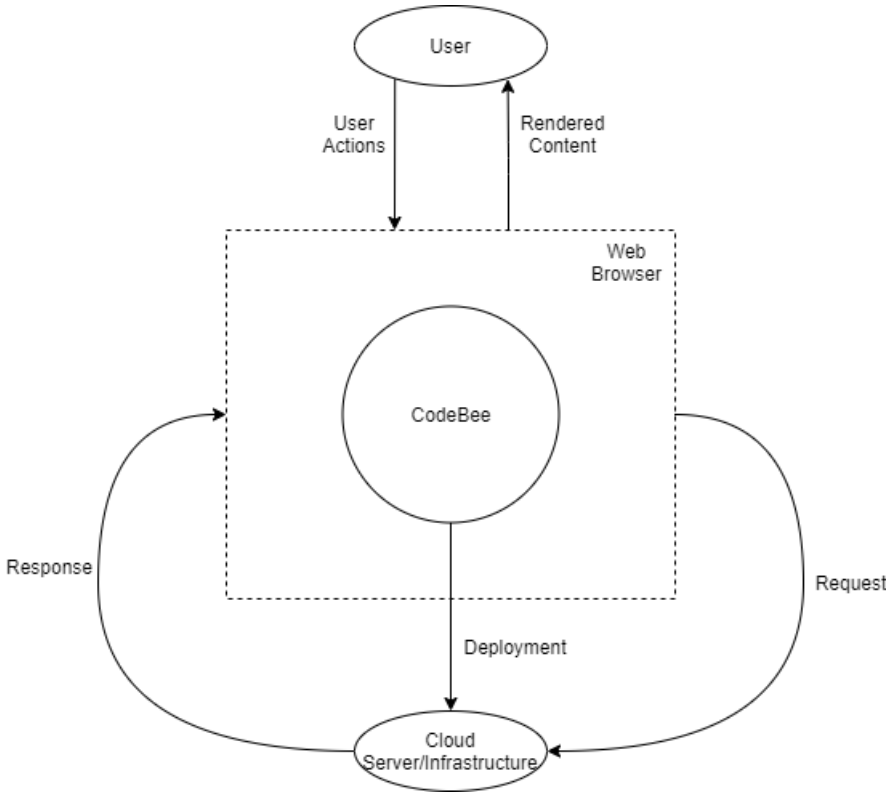


Figure 4: **Context Diagram**

CodeBee system will be run on a web browser and will be deployed on a Cloud Server. The user uses the web browser as an interface and performs user actions on the browser.

Then the web browser will make requests to the server according to user actions and get responses from it. The data acquired from the server will be rendered as output visualizations by the browser.

3.4 Dependency Diagram showing Functional Decomposition

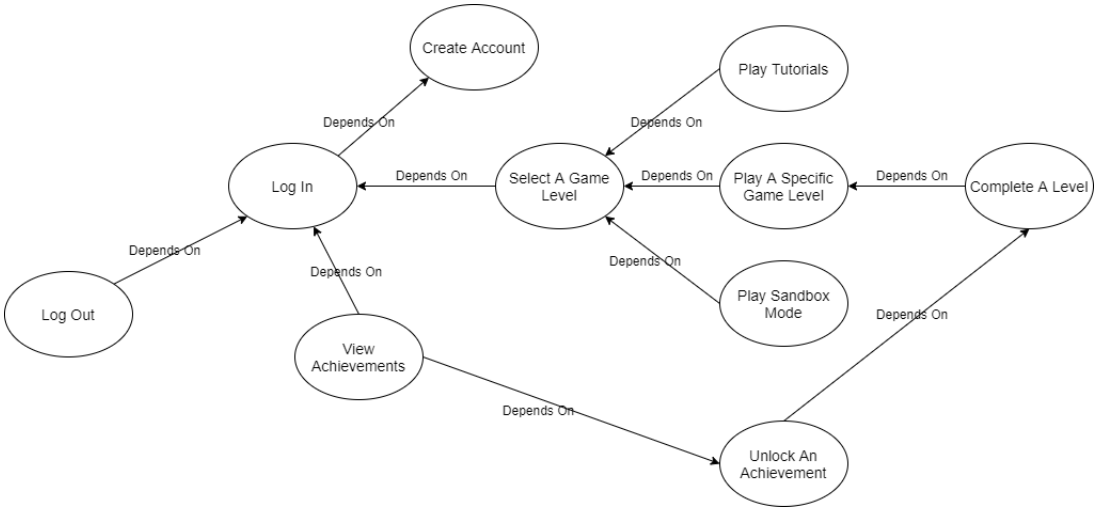


Figure 5: **Dependency Diagram**

In the dependency diagram, the decomposition of the functionalities of CodeBee system and the dependency among them are illustrated. The arrows in the diagram indicates dependency relationships. For example, A points to B means A depends on the completion of B.

4 Functional Requirements

4.1 User System

FR1.1	The system shall allow users to create an account consisting of a username and password.
Rationale	The system is designed to allow people to save their progress while logged in, so user annount creation must be possible.
Priority	1

FR1.2	The system shall allow users to use a login to use the system.
Rationale	Allows users to use their account with the system.
Priority	1

FR1.3	The system shall allow users to use the system without a login, as a guest user.
Rationale	Allows users without accounts to use the system.
Priority	2

FR1.4	The system shall record each logged-in users progress.
Rationale	To allow users to save progress, the progress must be tracked.
Priority	1

FR1.5	The system shall allow a guest user to progress for that session only.
Rationale	Users without an account should be able to advance through the system but since they lack an account, the progress should not be saved.
Priority	2

FR1.6	The system shall keep a record of each user who uses the system.
Rationale	Keeping a log of users can be useful for maintenance and analysis purposes.
Priority	4

FR1.7	The system shall not allow one user to access another user's information.
Rationale	Since private information is saved via system accounts, this information must be kept private.
Priority	1

FR1.8	The application shall allow the user to reset their password.
Rationale	Users can forget their password which might prevent them from accessing the application, so the user should be given an option to reset their password.
Priority	1

FR1.9	The application shall allow the user to log out of the application.
Rationale	In order to prevent unauthorized access to a user's account, the application should allow a user to log out of the application when they are not using it.
Priority	1

4.2 Level System

FR2.1	A level select screen must show the user all available levels
Rationale	The user must be able to select a level to play
Priority	1

FR2.2	The level system shall be able to load any arbitrary level that has already been unlocked.
Rationale	If a user is logged in, the user may choose to play any level they have unlocked.
Priority	1

FR2.3	The level system should not allow the user to load a locked level.
Rationale	If a user is logged in, the user should not be allowed to load a locked level as they might not have the prerequisite knowledge to attempt that level.
Priority	1

FR2.4	There must be multiple levels within the system.
Rationale	To facilitate user progression, there must be levels to progress through.
Priority	1

FR2.5	Excepting the first level, a level will become available to the user once that level's prerequisite level is completed.
Rationale	Levels will be unlocked sequentially, so every level besides the first will have prerequisite(s) and will therefore be locked.
Priority	2

FR2.6	A 'sandbox' level shall be available at all times that contains no instructions, no required output and has all code creation functionality unlocked.
Rationale	A sandbox level allows the user to use any knowledge they have gained in an unconstrained setting.
Priority	1

FR2.7	For a logged in user, each level will save the latest version of the user's solution.
Rationale	In the event the user wants to leave the level, the most recent user solution should be saved as progress.
Priority	3

FR2.8	The level select screen should show the user each level's name and a summary of the level contents before the user loads the level.
Rationale	The user should be able to know what the level is about before they load it.
Priority	3

4.3 Game System

FR3.1	The screen shall be separated into four primary components, the 'Instruction Box', 'Coding Area', 'Available Blocks' and 'Output'.
Rationale	For screen organization, the screen is separated based on the content in that space.
Priority	1

FR3.2	Within the Instruction Box will be utility function buttons.
Rationale	To control the game itself, buttons will be available to the user with various functions.
Priority	1

FR3.3	A utility function button will exist to allow the user to return to level select.
Rationale	The user must be able to leave a level and return to level select.
Priority	1

FR3.4	A utility function button will exist to allow the user to run their created program.
Rationale	The user must be able to run the program they created.
Priority	1

FR3.5	A utility function button will exist to allow the user to stop execution of their program.
Rationale	If the user encounters an endless loop, this button allows them to stop execution.
Priority	1

FR3.6	A utility function button will exist to allow the user to reset the content of their program to how it was at the start of the level.
Rationale	In case the user wants to start fresh, they should be able to reset the level to a blank state.
Priority	1

FR3.7	The instruction box shall contain any information required for the user to successfully complete the level.
Rationale	The instruction box tells the user what they must do to complete the level.
Priority	1

FR3.8	The instruction box shall not accept interaction from the user.
Rationale	The instruction box contains information that the user requires to complete the level, so the user should not be able to modify or remove the information.
Priority	1

FR3.9	The coding area shall provide the user with a space to create custom programs.
Rationale	To complete the level, the user requires a space to create a solution.
Priority	1

FR3.10	The coding area's contents shall be editable by the user.
Rationale	To create a solution, the user needs to be able to modify the solution.
Priority	1

FR3.11	The commands available to the user shall be restricted by the level they are on.
Rationale	To prevent overwhelming the player with commands they do not recognize, the available commands to the player will be restricted to the level they are on.
Priority	2

FR3.12	If the player has completed a level past the level they are playing, the player can choose to use the commands available originally in the level or the commands available in the later level.
Rationale	As the player progresses they will unlock more complicated and useful commands which can be used to solve previous solutions in new ways.
Priority	4

FR3.13	The commands available to the user will be shown in the Available Blocks area.
Rationale	The user may not have all commands available to them in a given level, so what is available must be shown.
Priority	2

FR3.14	The output section shall populate with the result of the user's code when the execute button (FR3.4) is pressed.
Rationale	The user needs a method to execute their program and retrieve the output provided.
Priority	1

FR3.15	Upon pressing the execute button (FR3.4) and output being generated, the game shall compare the contents of the output to the level's correct output to determine if the user has completed the level.
Rationale	For the level to be judged as completed or not, the user's solution output must be judged against the correct output.
Priority	1

FR3.16	The game shall inform the user that they have not successfully completed the level when they hit the execute button (FR3.4).
Rationale	The user should be informed that they haven't successfully completed the level and that some errors have been made.
Priority	1

FR3.17	The game shall prompt the user with options to try the level again or to go back to level select.
Rationale	The user should be informed that they haven't successfully completed the level and that some errors have been made.
Priority	1

FR3.18	Upon successful completion of a level, the user shall be given options to continue to the next level, stay on the current level or go back to the level select.
Rationale	Upon completion of a level, the user may wish to stop, go to the next level or continue on this level.
Priority	1

FR3.19	During execution of the user's program, pressing the stop button (FR3.5) will stop program execution and clear the output area.
Rationale	In the event that the user wishes to cancel running their program, the stop button will allow them to.
Priority	1

FR3.20	The application allows the user to save the progress they have made within a level.
Rationale	In order to avoid having to start the level from the beginning every time, and to avoid unnecessary repetition, the user will be allowed to save the amount of progress they have made within a level.
Priority	2

FR3.21	The application allows the user to reset the level to a saved state.
Rationale	In order to avoid having to start the level from the beginning every time, and to avoid unnecessary repetition, the user will be allowed to load the level to a saved state.
Priority	2

FR3.22	Upon pressing the execute button, the application informs the user that their solution is taking too long to run.
Rationale	The application should tell the user if their solution takes too long to run or times out.
Priority	3

FR3.23	The application shall allow the user to view solutions or hints to a level by spending in-game currency.
Rationale	If a user gets stuck with a level, they should be able to get some form of help in order to move to the next level.
Priority	3

4.4 Achievement System

FR4.1	After completing certain levels, the player shall unlock an achievement.
Rationale	Users should be rewarded for completing levels.
Priority	1

FR4.2	Unlocking of an achievement will cause a notification to show.
Rationale	The player should be notified that they have completed a feat within the game.
Priority	3

FR4.3	User shall be able to view their unlocked achievements in an achievement overview screen.
Rationale	The user should be able to see all their advancements.
Priority	2

FR4.4	After completing certain levels and fulfilling certain solution requirements, the player shall unlock an achievement.
Rationale	In addition to progression achievements, the player shall be rewarded by completing levels in ways that utilize advanced concepts for that level.
Priority	2

FR4.5	The user shall be able to view achievements that they can unlock along with the requirements for these achievements.
Rationale	The user should be able to see their own progress through the system by viewing their unlocked and unlocked achievements.
Priority	2

FR4.6	The user shall get points for unlocking achievements that they can use to purchase hints or solutions to levels in the game.
Rationale	If the user gets stuck in the game, they should have a way to get help through CodeBee.
Priority	3

FR4.7	The user shall get points for completing a predetermined set of achievements.
Rationale	To take advantage of the help system within CodeBee, the user must have some way of gaining points.
Priority	3

5 Non-functional Requirements

5.1 Look and Feel Requirements

5.1.1 Appearance Requirements

LF1.1.1	The product graphics shall be attractive to teenagers.
Rationale	Since the target users are teenagers aged 10-16, the graphics should be attractive to them to increase their interests in the product.

5.1.2 Style Requirements

LF1.2.1	The style of graphics shall be teenager-friendly and playful.
Rationale	Since the target users are teenagers aged 10-16, the graphics should not contain any information that is not suitable to be shown to teenagers, such as violence and pornography.

LF1.2.2	The layout of the product shall be more like a breakthrough game rather than a text compiler.
Rationale	This requirement enables the most important feature of CodeBee, one which is the graphical visualization of program execution.

5.2 Usability and Humanity Requirements

5.2.1 Ease of Use Requirements

UH2.1.1	The product shall be easy for teenagers with rudimentary computer skills to use.
Rationale	The users should have some experience with computers and web browsers before they use CodeBee since CodeBee highly depends on a computer web browser. Also, the operations should not be too hard for teenagers to perform.

UH2.1.2	All of the operations supported by the system shall be able to be done with pointing devices and keyboards.
Rationale	Pointing devices and keyboards are expected to be computer-associated devices that all households commonly have.

5.2.2 Learnability Requirements

UH2.2.1	Users shall be able to finish every level with the help of tips or instructions.
Rationale	The users might get lost in a certain game level. Therefore, tips or instructions would increase the learnability of the system.

5.2.3 Understandability Requirements

UH2.3.1	The text on every block shall be easy to understand by teenagers.
Rationale	Since the target users are teenagers, the text on blocks should be readable and understandable by teenagers.

UH2.3.2	The product shall use simple English for everything else that could be easily understood by teenagers.
Rationale	Since the target users are teenagers, the text and description other than the block text decorations should also be readable and understandable by teenagers.

5.3 Performance Requirements

5.3.1 Speed and Latency Requirements

PR3.1.1	The input delay should be within half a second.
Rationale	The time latency that users see the reflection of their input should not be too long.

PR3.1.2	The loading time of levels shall be less than two seconds.
Rationale	Since each level is expected to be a small program, the loading time should not take too long.

5.3.2 Safety-Critical Requirements

PR3.2.1	The product must not display intermittently flashing light which might cause photosensitive epilepsy(PSE).
Rationale	The product shall not cause negative impact on the users' health.

5.3.3 Reliability and Availability Requirements

PR3.3.1	The product shall be able to handle any infrastructure failure in a robust manner.
Rationale	To ensure the product has some degree of robustness.

PR3.3.2	The service of the product shall be available all the time once the product is launched.
Rationale	To ensure the product has some degree of availability.

5.3.4 Robustness or Fault-Tolerance Requirements

PR3.4.1	In case that the users cannot access the server, an error page should be displayed.
Rationale	Error messages should be displayed if there is a disconnection. This follows the UI principles.

PR3.4.2	The server shall be able to store users' state when then disconnected from the server.
Rationale	To ensure the users have their progress saved before a disconnection. A work lost due to disconnection might decrease a user's personal satisfaction.

5.3.5 Capacity Requirements

CR3.5.1	The product shall be able to handle at least 100 simultaneous users in the first month since its publication.
Rationale	We estimate that the users in the first month are going to be 10-100. Therefore, the requirement need to guarantee the upper bound of user size should be capable.

5.4 Operational and Environmental Requirements

5.4.1 Expected Physical Environment

OE4.1.1	Operations of the product shall be able to be completed by teenagers with a basic set of computers and a pre-installed browser.
Rationale	The users should a computer and associated devices, and a web browser to use CodeBee.

5.4.2 Requirements for Interfacing with Adjacent Systems

OE4.2.1	The product shall support the latest released version of browser.
Rationale	Since the users might have different versions of browsers, CodeBee should operate properly on them.

5.4.3 Release Requirements

OR4.3.1	The released updates shall not cause the previous levels to fail, or cause progress to be reset.
Rationale	Each release should be compatible with the last or the previous version should be capable of being migrated to the new system to prevent loss of user progress.

5.5 Maintainability and Probability Requirements

5.5.1 Maintenance Requirements

MP5.1.1	The application shall be deployed and hosted on cloud servers in order to make the application easier to maintain.
Rationale	Cloud servers have many advantages for web applications, such as scalability and efficient recovery. These advantages increase maintainability.

5.5.2 Portability Requirements

MP5.2.1	The interface of the product is expected to scale according to the screen size, resolution and aspect ratio of the device.
Rationale	To satisfy different users' device screen size.

MP5.2.2	The interface of the product is expected to be compatible with various input methods.
Rationale	To satisfy different users' input methods, such as touchscreen, mouse, and keyboard.

5.6 Security Requirements

5.6.1 Access Requirements

SR6.1.1	The system shall restrict one user's access to the data of other users.
Rationale	To ensure the stability of the system and also to ensure the privacy of users.

SR6.1.2	The system shall not allow users to see the correct answers before they finish unless they use the help feature (FR4.7).
Rationale	To ensure the user will have a chance to think and learn.

5.6.2 Integrity Requirements

SR6.2.1	The product shall prohibit any unauthorized modification to the data stored online.
Rationale	To ensure data integrity of the system.

5.6.3 Privacy Requirements

SR6.3.1	The system shall not collect any user information without permission.
Rationale	To ensure the privacy of the users.

SR6.3.2	The product shall inform users of the change in privacy policies.
Rationale	To keep the users in control and let them be clear with the privacy policies.

5.6.4 Immunity Requirements

SR6.4.1	The password shall be encrypted before the transaction across the client and the server.
Rationale	To increase the system stability and protect users' privacy.

5.7 Cultural and Legal Requirements

5.7.1 Cultural Requirements

CL7.1.1	The interface or the setting of the levels shall respect all cultures. Any offensive elements shall not be used.
Rationale	To ensure the product is suitable to all cultural groups.

5.7.2 Compliance Requirements

CL7.2.1	The product shall comply with Copyright and Intellectual Property Law.
Rationale	To ensure the product will not cause legal offense.

5.7.3 Standards Requirements

CL7.3.1	The product shall comply with the protection standards made for teenagers, such as the Children's Online Privacy Protection Act(COPPA).
Rationale	To ensure the product is suitable for teenagers to use and meets the standard.