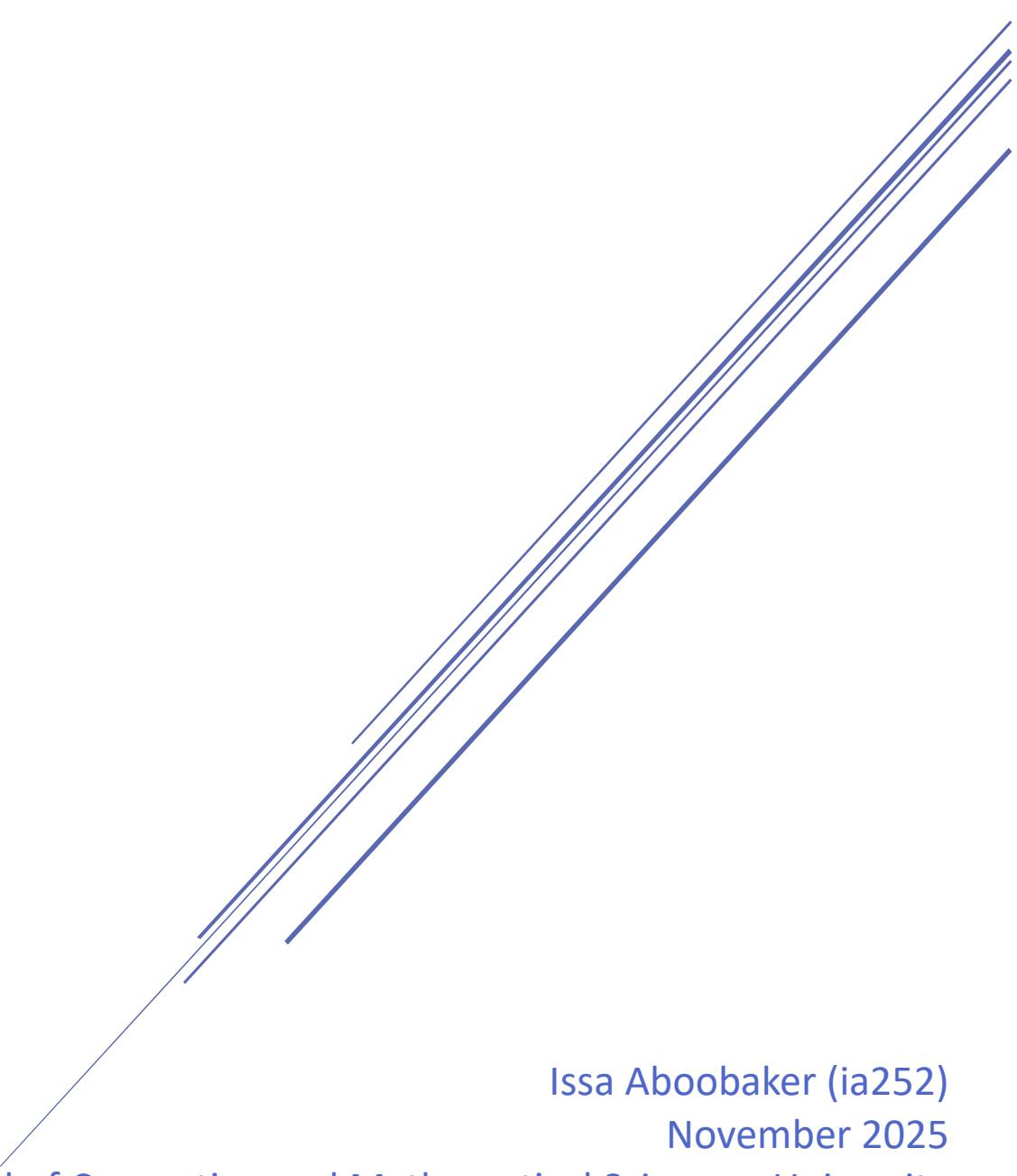


INTERIM REPORT

CO3201 Computer Science Project - Virtual Vending
Machine Webapp



Issa Aboobaker (ia252)
November 2025
School of Computing and Mathematical Sciences, University
of Leicester

Declaration

All sentences or passages quoted in this report, or computer code of any form whatsoever used and/or submitted at any stages, which are taken from other people's work have been specifically acknowledged by clear citation of the source, specifying author, work, date and page(s).

Any part of my own written work, or software coding, which is substantially based upon other people's work, is duly accompanied by clear citation of the source, specifying author, work, date and page(s).

I understand that failure to do this amounts to plagiarism and will be considered grounds for failure in this module and the degree examination as a whole.

Name: Issa Aboobaker

Signed:

A handwritten signature in black ink, appearing to read "Issa".

Table of Contents

1.	Aims and Objectives.....	3
1.1	Overview	3
1.2	Aims	3
1.3	Objectives	4
1.4	Challenges and Originality	5
2.	Survey of Literature.....	6
3.	Requirements.....	9
3.1	Functional Requirements.....	9
3.2	Non-Functional Requirements	10
3.3	Project Constraints.....	11
4.	Outline of Specification and Design.....	12
5.	Planning and Timescales.....	15
	Semester 1	15
	Semester 2	16
	Gantt Chart.....	18
6.	References.....	19

1. Aims and Objectives

1.1 Overview

My project idea is to create a virtual vending machine webapp. This will simulate a real vending machine and allow users to choose specific items to vend and then insert coins into the machine to pay for their items. They will be able to choose multiple items which they can pay for and vend all at once. If necessary, change will be dispensed from the machine. The machine will automatically update its stock when running low.

1.2 Aims

The aims refer to what I intend to do and achieve throughout the development of the project. They aren't directly measurable, but they outline what I need to do and what to include in the project.

- To design and develop an interactive web application that simulates the exact functionality of a real-life vending machine
- To create a user interface that is attractive to users and allows them to seamlessly use the app and have an enjoyable experience
- To enhance my understanding of web development concepts: MVC architecture, Spring Boot, web page design (HTML/JSP, CSS, JavaScript), database integration.
- To create a project containing parts that may be reusable – some concepts or designs might be helpful to use in future projects.
- To design a software that is scalable – the design/layout of the software should allow new features to be easily added in the future
- To improve my coding and project management skills: Java coding, mathematical skills/concepts, logical decision making, data structures

and algorithms, code robustness, maintainability, code structure, file organization, time management, deadline setting, documentation, design ideas, testing, evaluation.

1.3 Objectives

The objectives refer to the steps that I need to take to ensure that the project is a success. They are defined by my own targets and goals and are based on the aims. They are measurable so I can always refer to them during the development stage to see if I am on track to achieving all of them.

1. Conduct research/testing on existing similar systems and use this to decide features/requirements
2. Present these requirements in detail in this report
3. Make a time plan for the entire development process (shown later in this report)
4. Create the web application using Spring MVC Framework
5. Implement backend functionality of the app using Java classes
6. Design the webpage layout using JSP/HTML
7. Style the webpage using CSS to make it look appealing to the user
8. Enhance the webpage using JavaScript to make it dynamic and interactive and create a high-level user interface
9. Ensure code is well structured and documented, using comments everywhere throughout
10. Perform thorough unit and UI testing throughout the development process and perform end user testing at the end

1.4 Challenges and Originality

The problem I'm trying to solve is to create an effective and user-friendly online simulation of a vending machine. It is intended to function exactly like a real-life vending machine to give an accurate simulation. This can allow many real-life applications such as providing convenience and quick access to online products which allow users to explore and make purchases remotely. It can also serve as a digital prototype that businesses can use to test vending machine functionality before they invest in physical machines. Finally, it can be used educationally to train individuals in how to use vending machines which they may come across in lots of different places:

- Schools, Colleges and Universities
- Sports centres and gyms
- Food places, restaurants, cafes
- Shopping malls
- Stadiums
- Hospitals
- Train stations, bus stations

Virtual vending machines already exist. I intend for my work to be an improvement on these existing systems. Firstly, a big focus will be on the frontend development. Many of the existing systems have poor quality user interfaces which have things such as slow interactions and basic appearances. I intend to design my user interface to operate smoothly and to look appealing to the user. I also intend to add extra features to improve the functionality of the vending machine, for example my proposed system will allow the user to select multiple items and then pay for them together as opposed to existing systems forcing users to vend items one at a time. Some other added features for my system include: stock management, the option to print a receipt and dispensing change. With all of this I am trying to develop a proper and accurate simulation of a real vending machine so it can be used to solve the intended problem and can be implemented for real life applications as listed before.

2. Survey of Literature

To successfully carry out my virtual vending machine project, I have conducted thorough background research on key topics including vending machine operations, existing virtual vending machine simulations and user interface design principles. I have summarised my findings below.

Firstly, I researched how traditional vending machines function, including inventory management and item dispensing processes. Understanding these operational aspects is crucial for replicating realistic functionality in the virtual machine. From the source [1] I was able to understand that vending machines use sensors to track stock levels of items in “real time”. Advanced technology means that sensors can track every item to allow operators to know “exactly when supplies are running low and what needs to be replenished”. This is helpful to understand how to manage stock in my project. As the machine is going to be virtual there is no need for physical sensors but there needs to be some form of logic that can act as a sensor and detect when items are running low and instantly recognise that the stock needs to be updated. From the next source [2] I understand that typically, vending machine items are held in place by “electric coils” which are “activated” when payment has been completed and allows the item to drop down for the customer to claim it. Newer machines have mechanical arms to pick up the product and drop it into the collection tray to prevent damaging the product. In my project, there will be no requirement for any such hardware as it is virtual, but I may decide to include some sort of animation to simulate the dropping or movement of items to add realistic effects.

My next topic of research was the most important. I found and tested other existing virtual vending machine simulations to see which features are useful and things that are missing so I can decide upon my development ideas. The first one I tested [3] was quite a similar system to my proposed idea. The vending machine visually displayed the items and allowed me to insert coins and dynamically updated my balance as I did. It allowed me to select an item to vend and even visualised the item dropping out of the machine and updating my remaining balance on the display. However, the display was quite blurry and

unclear and the website overall operated quite slow and laggy. I intend for my system to have similar functionality but much better performance and to operate smoothly. Another system I tested [4] was interesting to use but the features were different. It involved picking a combination of different shapes, colours and sizes and the machine would accurately vend this. It was engaging to test out and the performance level was quite high. However, it didn't contain features of a conventional vending machine so it lacked the realistic elements and can't be used to model a proper vending machine in the way I want to. The final system [5] that was helpful to test had a clear and appealing user interface. It also had some complex functionality. It allowed me to view the price of an item by entering its code into a keypad. It then allowed me to press an enter button which then prompted me to start entering coins. As I did this, it dynamically reduced the total until I had paid the required amount. The system also had some animations with the item after it had been vended. For my project I may want to consider adding the keypad feature to select items rather than just clicking them. The payment process from this system is also something impressive for me to consider in my project. The only real criticism of this system is that it is very childish and playful and doesn't emulate the realistic simulation of a vending machine that I am aiming for. Overall, from testing other existing systems, I have concluded that they have some desirable features that I may want to include in my project but the main thing to focus on to make my project unique and an improvement on the rest, is the user interface. I intend to make the machine look and act realistic to give the best possible simulation.

Finally, the research paper [6] helped me to understand that three essential components for user interface design are:

- “1. Place the user in control.
- 2. Reduce the user's memory load.
- 3. Make the interface consistent.”

These principles emphasise the importance of creating a user interface that empowers the user by giving them control over interactions and reduces what they must know and remember to be able to utilise the system to its max. Consistency in design helps the user navigate the interface more efficiently.

Evaluating my research, the existing systems were the most helpful as they gave me an understanding of which features and parts of the project were most important to include but also what improvements and additions to consider. The research paper [6] was also particularly helpful as it gave me an insight into what is important in developing a high-quality user interface.

3. Requirements

After reviewing the objectives and conducting research into other similar systems, I have come up with the requirements for the project. These have been grouped into 3 categories: Functional Requirements, Non-Functional Requirements and Project Constraints.

Functional Requirements are the specific features and functionalities that the application must provide to the user.

Non-Functional Requirements refer to the quality and technical requirements of the application.

Project Constraints include the restrictions and deadlines which the project must be completed within.

3.1 Functional Requirements

1. Product display:

- Show all available products with details (name, price, quantity)

2. Product Selection:

- Allow users to select which product(s) they want and add to their cart

3. Cart Management:

- Calculate the total cost of the cart and update this when necessary
- Allows user to add and remove things from cart

4. Payment Processing:

- Allows user to choose coins to input into machine
- Automatically deducts coin value from total left to pay

5. Change dispensing:

- Dispenses correct change in coins if necessary

6. Receipt Printing:

- Allows user to have a receipt printed out for each transaction if they choose to

7. Stock Management:

- Automatically updates stock in the vending machine when running low

8. Interactivity:

- Application contains JavaScript to allow interactivity for the user

9. Interface:

- Aesthetically pleasing design and layout of vending machine for user satisfaction (CSS)

10. Feedback:

- App shows messages and prompts when a user has performed an item to keep them informed and engaged (e.g. “Payment has been completed”, or “Item added to cart”)

3.2 Non-Functional Requirements

11. Performance:

- The webpage should be loaded within a few seconds and so should any dynamic updates

12. Scalability:

- The code design and structure should allow for easy addition of new features/products

13. Error Handling:

- The code should be equipped to deal with basic errors such as items out of stock or invalid selections

14. Maintainability:

- Code should be organized well and have documentation/comments

15. Testing:

- Testing must be carried out to ensure all features and the interface are working properly

3.3 Project Constraints

16. Technologies used:

- Backend: Spring Boot, Java
- Frontend: HTML/JSP, CSS, JavaScript
- Build Tool: Gradle

17. Timeline:

- The project development process must follow the defined time plan (in this report)

18. Documentation:

- A project log must be made and constantly updated throughout development

19. Development Approach:

- Project must follow MVC architecture

4. Outline of Specification and Design

The vending machine simulation is designed with a Spring Boot backend and a JSP/HTML/CSS/JavaScript frontend for enabling product management, transaction processing, and user interaction. The key components and technical details are described below.

1. Backend

The backend will be developed using Spring MVC. The main elements are:

- Product Management:
The system includes a Product class that defines attributes like name, price, and stock level. A ProductService handles all product-related operations, such as adding, removing, and updating products. Product data is stored in a database (mySQL) with tables for products, and a ProductRepository for data handling.
- Transaction Handling:
A Transaction class manages details like purchased items, total cost, and payment sums. The transaction logic includes:
 - Payment Validation: Ensures the user has provided sufficient funds for the transaction.
 - Change Calculation: Calculates the exact change to be dispensed based on payment given.
 - Stock Updates: Subtracts the purchased item quantities from the database after successful transactions.
- Receipt Generation:
The system generates formatted receipts that include the list of purchased items, their prices, the total amount, and change if applicable. These can then be optionally printed by the user.

2. Frontend

The frontend provides a user-friendly interface for interacting with the vending machine.

- User Interface:
Built using JSP/HTML, the UI displays products in the machine, showing details such as name, price, and stock. Users can select items and view them in their cart.
- Dynamic Features:
JavaScript dynamically updates the cart, calculates the total cost, and reflects stock changes in real-time. AJAX could potentially be used to make backend calls without reloading the page.
- Styling:
The UI is styled with CSS, ensuring user engagement and satisfaction.

3. Integration

The backend and frontend are connected through RESTful APIs. These APIs handle requests such as fetching product data, processing transactions, and updating stock levels.

- Example API endpoints:
 - GET /products – Retrieves product stock level.
 - POST /transactions – Completes a purchase.

4. Algorithms and Data Structures

- Algorithms:
 - Payment Validation: Ensures user payment matches or exceeds the total cost.
 - Change Calculation: Identifies the correct change value and breaks down the change into the coins.
 - Stock Management: Checks product availability and updates stock levels after purchases.
- Data Structures:
 - Classes: Product and Transaction are used to model stock levels and user transactions.
 - Database: A database stores products and transactions. Tables are queried using Spring repository interfaces.
 - Arrays and lists may need to be used for temporary data storage.

To summarise, the system is a well-structured vending machine simulation that integrates a robust Spring Boot backend with a dynamic frontend. It utilises algorithms for payment validation and stock updates, uses a database for secure data storage and offers an immersive user experience through a well-designed UI.

5. Planning and Timescales

I have shown my time plan in two tables, one for each semester. I have split up the project time into stage for each part of the project to make it more organized and manageable. At the end of each stage there is a “**Checkpoint**” to allow me to go over all the progress in each phase and check/test everything. The dates may not turn out to be exactly accurate as some tasks may need more/less time than expected. Also, there is some overlap between the semesters due to the Christmas break and I can’t split the task dates midway through a task, so the semester dates aren’t exact but are approximate. I have left a large chunk of time at the end before the submission deadline so in case anything doesn’t get done, I can go back to complete it. This allows me to leave out difficult or time-consuming parts so I can continue with the time plan and revisit things at the end.

Semester 1

Stage	Task	Start Date	End Date	Status
1. Research & Planning				
	Research existing vending machine simulations to identify features and user expectations.	22/10/2024	07/11/2024	Completed
	Define project aims and objectives	08/11/2024	10/11/2024	Completed
	Define system requirements	11/11/2024	13/11/2024	Completed
	Develop a detailed project time plan (this)	14/11/2024	16/11/2024	Completed
Checkpoint 1		17/11/2024	19/11/2024	
2. System Design				
	Design the system architecture using Spring MVC Framework.	20/11/2024	24/11/2024	In Progress

	Plan the database layout to manage product stock	20/11/2024	26/11/2024	In Progress
	Create wireframes for the user interface.	20/11/2024	30/11/2024	In Progress
Checkpoint 2		01/12/2024	03/12/2024	
3. Backend Development				
3.1 Set Up Backend	Create a Spring Boot project using Gradle.	21/11/2024	03/12/2024	In Progress
	Add dependencies for Spring MVC and other required libraries.	21/11/2024	05/12/2024	In Progress

Semester 2

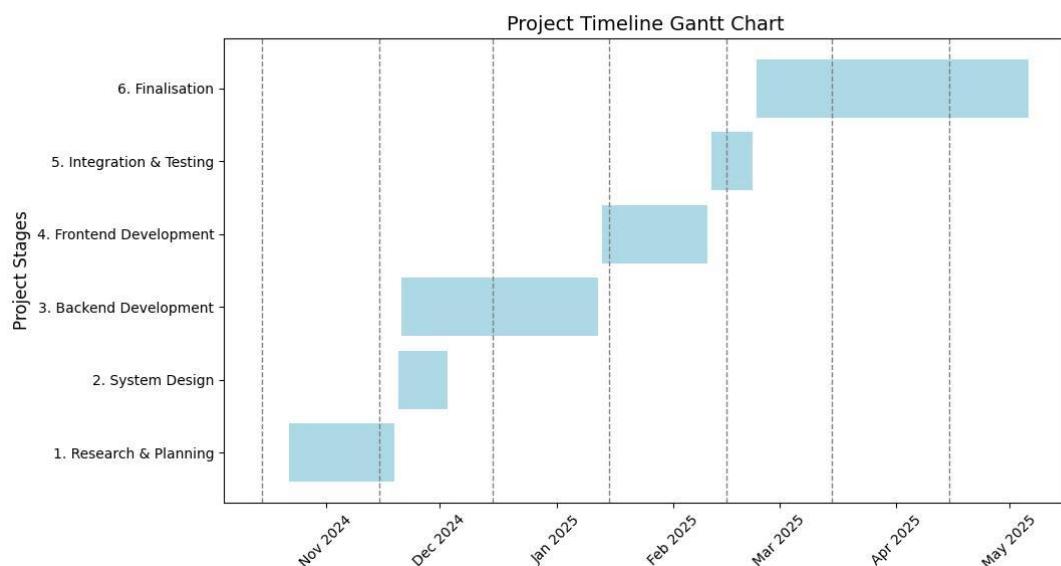
Stage	Task	Start Date	End Date	Status
3.2 Manage Products (Stock)	Create a Product class with attributes: name, price, and stock level.	06/12/2024	08/12/2024	Not Started
	Set up a service for product operations	09/12/2024	11/12/2024	Not Started
	Implement the database connection to store product information.	12/12/2024	14/12/2024	Not Started
	Add a method to update stock levels when they become low.	15/12/2024	17/12/2024	Not Started
3.3 Handle Transactions	Create a Transaction class to store details like purchase items and total price.	18/12/2024	20/12/2024	Not Started
	Code the logic for selecting multiple products and calculating the total cost.	21/12/2024	23/12/2024	Not Started
	Add a method to check if the user's payment matches the total price (or if it exceeds it).	24/12/2024	26/12/2024	Not Started

	Code the logic to dispense change if required.	27/12/2024	29/12/2024	Not Started
3.4 Generate Receipts	Create a method to generate a receipt with the purchased items, prices, total and change (if there is).	30/12/2024	31/12/2024	Not Started
	Format the receipt for printing.	01/01/2025	02/01/2025	Not Started
3.5 Testing & Debugging	Perform unit tests for product management features.	03/01/2025	05/01/2025	Not Started
	Perform unit tests for transaction logic (e.g. payment, change dispensing).	06/01/2025	08/01/2025	Not Started
	Test backend code to ensure it works as expected.	09/01/2025	11/01/2025	Not Started
Checkpoint 3		11/01/2025	12/01/2025	
4. Frontend Development				
4.1 Webpage Layout	Design the basic webpage layout in JSP/HTML.	13/01/2025	16/01/2025	Not Started
	Create individual components for product display (e.g. product name, price, and stock).	17/01/2025	20/01/2025	Not Started
4.2 Cart Functionality	Implement a cart system and add JavaScript to dynamically show selected items.	21/01/2025	25/01/2025	Not Started
4.3 Styling	Add styling using CSS to ensure the layout looks appealing.	26/01/2025	29/01/2025	Not Started
4.4 Dynamic Features	Add more JavaScript for interactivity.	30/01/2025	04/02/2025	Not Started
	Prepare for interview	03/02/2025	17/02/2025	
4.5 Frontend Testing	Test all frontend components for functionality.	05/02/2025	08/02/2025	Not Started
Checkpoint 4		09/02/2025	10/02/2025	
5. Integration & Testing				

	Ensure connection between backend and frontend.	11/02/2025	13/02/2025	Not Started
	Conduct integration testing for complete functionality.	14/02/2025	16/02/2025	Not Started
	Fix any identified bugs or issues.	17/02/2025	19/02/2025	Not Started
Checkpoint 5		20/02/2025	22/02/2025	
6. Finalisation				
	Complete project documentation: user manual and project log.	23/02/2025	01/03/2025	Not Started
	Revisit unfinished tasks or add additional features/improvements.	02/03/2025	29/03/2025	Not Started
	Final checks and testing.	30/03/2025	31/03/2025	Not Started
	Dissertation.	01/04/2025	01/05/2025	
	Prepare for submission.	01/04/2025	01/05/2025	
	Viva.	01/04/2025	06/05/2025	

Gantt Chart

As most details have been shown in the tables, the Gantt chart has been simplified to show the main stages of development and their time frames.



6. References

Relevant citations of sources mentioned in the *Survey of Literature* section:

- [1] Vending-Machines.ie, "How do vending machines manage inventory and supply chain?," LinkedIn, Nov. 22, 2024. [Online]. Available: <https://www.linkedin.com/pulse/how-do-vending-machines-manage-inventory-supply-chain-zed2c/>.
- [2] NUVending, "How do vending machines work?," NU Vending, [Online]. Available: <https://www.nuvending.co.uk/news/how-do-vending-machines-work/#:~:text=Typically%2C%20vending%20machines%20have%20rows,pickers%20to%20retrieve%20the%20product>.
- [3] PlayHop, "Vending Machine Simulator," PlayHop, [Online]. Available: <https://playhop.com/app/339874>.
- [4] D. Roberts, "VIRTUAL VENDING MACHINE: An interactive vending machine. Weekly game design project," Dennis Roberts, [Online]. Available: <https://dennisroberts.com/filter/GAME/VIRTUAL-VENDING-MACHINE>.
- [5] ArcadeSpot, "Surprise Eggs Vending Machine," ArcadeSpot, [Online]. Available: <https://arcadespot.com/game/surprise-eggs-vending-machine/>.
- [6] S. Sridevi, "User Interface Design," *Int. J. Comput. Sci. Inf. Technol. Res.*, vol. 2, no. 2, pp. 415-426, Apr.-Jun. 2014. [Online]. Available: www.researchpublish.com.