

Software Design Document

Project Anglesea

Team Members

Name	SID	Email
Hugo Cawsey (HC)	47293241	hugo.cawsey@students.mq.edu.au
Md Tanvir Mahtab (M)	45822832	mdtanvir.mahtab@students.mq.edu.au
Blake Kouzeleas (B)	47217766	blake.kouzeleas@students.mq.edu.au
Haiderabbas Momin (HM)	47297379	haiderabbas.momin1@students.mq.edu.au

Change Log

Date	Change	Author(s)	Members Agreed
16/10/23	Added Team Member table and Change log	HC	HC, M, B, HM
16/10/23	Started Vision Statement	HC	HC, M, B, HM
16/10/23	Finished Vision Statement	HC	HC, M, B, HM
16/10/23	Started Requirements Traceability Matrix	HM	HC, M, B, HM
21/10/23	Started System Design Section	HC	HC, M, B, HM
23/10/23	Started Project Management Section	B	HC, M, B, HM
23/10/23	Finished System Design Section	HC	HC, M, B, HM
25/10/23	Started Test Specifications	HM	HC, M, B, HM

Date	Change	Author(s)	Members Agreed
25/10/23	Started Analysis and Design	M	HC, M, B, HM
25/10/23	Finished Test Specifications	HM	HC, M, B, HM
25/10/23	Started Data Definitions Table	HC	HC, M, B, HM
26/10/23	Finished Data Definitions Table	HC	HC, M, B, HM
27/10/23	Finished Requirements Traceability Matrix	HM	HC, M, B, HM
27/10/23	Finished Project Management Section	B	HC, M, B, HM
28/10/23	Finished Analysis and Design	M	HC, M, B, HM
29/10/23	Added Descriptions for Milestones in Project Management Section	B	HC, M, B, HM
1/11/23	Updated Appendicies	HC	HC, M, B, HM
1/11/23	Updated Data Definitions	HC	HC, M, B, HM
2/11/23	Updated System Overview	HC	HC, M, B, HM
3/11/23	Updated Project Management	B	HC, M, B, HM
4/11/23	Added Outlook	HC	HC, M, B, HM
4/11/23	Changed Test Specification Fromat	HM	HC, M, B, HM

Date	Change	Author(s)	Members Agreed
4/11/23	Updated Project Management	B	HC, M, B, HM
4/11/23	Final Touche-ups of Doc	HC	HC, M, B, HM

Vision Statement

Our vision at project Anglesea is a sustainable coffee culture that seamlessly unites all coffee drinkers with local businesses. We strive to redefine coffee consumption by offering a subscription-based model that allows users to enjoy their coffee in reusable cups at participating local shops without the need for ownership or cleaning of cups. Our innovative app streamlines cup management for local coffee shops and increases beneficial coffee habits for users that minimise waste, supporting an eco-friendly landscape.

We aim to start with a local community, encompassing a larger area as we start to become established. Starting our pilot project with a selected university, we can adapt our project for any change to come. Through our technology, we envision a future where coffee drinkers undergo a care-free and easy drinking experience, thriving from local economies and business, whilst encouraging a greener future by reducing single-use plastics and welcoming sustainable choices.

System Design Document

System architecture

Project Anglesea's Coffee Cup Subscription and Management System (CCSMS) will feature the utilisation of a three-tier based system architecture. This will allow us to split the project into three main subject components, the Application Layer, System Layer, and the Data Layer. These three distinct layers will encapsulate the basis of the system and was chosen to improve scalability, reliability, and security.

Application Layer The Application Layer in the CCSMS serves as a link between the user interface and the data layer, offering a range of functionalities and services to users, coffee shops, and system administrators. This layer provides various components, including the user interface, application logic, and authentication mechanisms. Users will interact with the system through this layer, accessing features like subscription management, personal coffee cup tracking, loyalty program participations and many more features.

This layer will also be designed to ensure a secure and efficient time for the user experience, through numerous functional and aesthetic testing, as well as

facilitating communication with the Data Layer for data retrieval and updates. Overall this Layer is crucial and a core components of the CCSMS architecture, providing the backbone for user interface and smooth system experiences.

System Layer *Microservices Architecture:* Building the CCSMS using a microservices architecture will provide efficient development and scalability, allowing for the allocation of resources where needed. Each major functional aspect will be managed by a separate microservice. This modularity supports each service to have its data store, optimised for specific needs. The system to be structured into small, independent services, each focused on specific functions, including: - User Management - Subscription Handling - Cup Tracking - Administration Access - Systems Handling

For example, scaling the services handling cup tracking or coffee shop integration during peak time without disrupting the entire application will be a beneficial and effective use of this architecture.

Inferring that for the CCSMS, adopting the microservices architecture will allow a flexible, scalable and maintainable system that can adapt to the change in requirements at the time, handling the complexity of managing subscriptions and coffee cup tracking efficiently. However, this architecture does require careful consideration and planning to ensure effective communication, data consistency and system integrity throughout.

Data Layer The Data Layer as one of the core components of the CCSMS architecture is responsible for storing, managing, and providing access to the system's data, ensuring data integrity, real-time access to relevant information. This layer is divided into two primary categories, each serving specific needs within the CCSMS architecture: the Relational Database and the NoSQL Database.

The Relation Database serves as the repository for more structured data whilst the NoSQL Database is the engine behind reliable fast and efficient data transfers for our system such as the Coffee Cup real-time Tracking. Each sections were chosen for the benefits over other options and are discussed further in the 'Storage / Persistent Data Strategy sections'. Moreover, having secure Data Encryption, whilst being a non-functional requirement, was a persuading factor when defining our project.

Storage / Persistent Data Strategy

The storage and persistent data strategy is a critical component of our CCSMS and its functionalities. It involves the decision on how data is stored, managed and accessed throughout the system and to its users.

Relational Database: A relational database management system such as MySQL, will be used for storing structured data that is essential for the system.

This includes information related to user profiles, coffee shop information, and other structured data.

We chose a relational database system as it is ideal for maintaining data consistency and enforcing data integrity, support complex queries for report and analytics (i.e. User Subscription Analytics).

NoSQL Database: Since the project focuses on real-time tracking of coffee cup usage, it is essential for a database to have quick insertion and retrieval of data for such functions.

NoSQL is an appropriate choice for real-time tracking, as these databases are well-suited for high input scenarios where it is essential for data to be quickly inserted, updated and retrieved on demand. For our CCSMS, this database will store data relating to individual coffee cups, their locations, usage history and further associated metadata. The NoSQL database, has a schemaless structure, allowing for flexibility in handling rapidly changing data, further benefiting its use.

Data Encryption: Since our CCSMS will be handling secure and sensitive information like user profiles and their associated payment details, the security of this data is of utmost importance. Both through rest and in transit, data encryption should and will be implemented. This will include SSL/TLS (secure sockets layer and transport layer security) encryption when communicating data between the system and databases, establishing a secure and safe connection.

Using these security and encryption measure should protect against data breaches and unauthorised access, furthermore, aligning with Non-Functional Requirement 4.

Noteworthy trade-offs and choices

Whilst all system design decisions are made to most benefit the system, each decision will have associated trade-offs that may affect other areas of the overall system. The following are Noteworthy trade-offs and choices.

System Architecture: *Choice:* Microservices Architecture *Trade-offs:* The need for inter-service communication, version management and monitoring tools become more prominent the more this architecture is used. Scaling the project larger will require management of more and more services, requiring a balance between flexibility and complexity.

Security over Convenience: *Choice:* Encryption of sensitive data at rest and transit to enhance security. *Trade-offs:* Favouring encryption introduces heavier computational overhead, which may lead to impacting system performance, however, may not be noticeable at smaller levels. The need for a balance between security and efficiency becomes more prominent when deciding the level of data protection for user's sensitive information.

NoSQL DataBase: *Choice:* Implementing NoSQL databases for real-time tracking of coffee cup usage and related functions. *Trade-offs:* While NoSQL databases provide high-speed data insertion, updating and retrieval, they lack the strong consistency and query capabilities of SQL databases. Where different components within the CCSMS require a more structured database, NoSQL would impact functionality. However, for the case of live-tracking and other real-time data, NoSQL databases work effectively.

Concurrent processes (if any) and how they will be coordinated

A Concurrent process refers to multiple tasks or activities that can be executed simultaneously, taking place concurrently in the system. Effective coordination of concurrent processes are essential for a system to maintain data integrity, prevent conflicting functions, and to provide an efficient experience for the user. With the CCSMS Project, concurrent processes and the coordination of said processes will play a vital role in the success of the system's functionality, reflecting the system's responsiveness to higher demand. Concurrent processes for the CCSMS would include:

Coffee Cup Borrowing and Returning:

Processes: - User borrowing a cup - User returning a cup - Coffee shop owners managing cup availability

Coordination: Implement a transaction queue that captures the borrowing and returning of cups simultaneously. Each request is processed in a linear manner, i.e. if a cup is borrowed and another is returned, the system will first update the stock from the borrowed cup then the returned cup. This will prevent sudden rises or falls in cup availability in the case of build up. Additionally, coffee shop owners will be able to regulate their stock / cup availability for users manually in the case of broken cups or other faults.

Subscription Management and Real-Time Tracking:

Processes: - User subscribes to the coffee cup plan - Real-time updates of coffee cup and shop availability

Coordination: Upon selecting and finalising a subscription plan, user accounts will be updated with their selection plan, granting them access to real-time tracking of coffee shops and cups. The system will check for updates in the case of a users subscription changes or cancellations, this will then coordinate with the Real-time tracking to stop updates for live tracking in the case of subscription cancellation.

Package Diagram: *Figure 1.0*

Data Definitions

User Data

<i>Name Field</i>	<i>Type</i>	<i>Meaning</i>	<i>Example</i>
User ID	Hexadecimal	The unique identifier for each user	0x5c4a
User Name	String	The name stored in the user profile	John Smith
User Password	String	The password for logging into a user profile	JSPassword21
User Email	String	The email associated with the users account	JohnSmith@angleseamail.com
Payment Details	String	The users card number	1234 5678 9123
Subscription Active	Boolean	Subscription status will show true if the user has an active subscription	True / False
Loyalty Program	String	The name of the loyalty program the user has registered too	Esc Cafe Loyalty Program

Cup Data

<i>Name Field</i>	<i>Type</i>	<i>Meaning</i>	<i>Example</i>
Cup ID	Hexadecimal	The unique identifier for each cup	0xfe253e
Cup Location	Float	Location of cups last drop-off / pickup (where the RFID Tag was last scanned)	-0.4579876, 2.3456786
Cup Availability	Boolean	Will show true or false depending on if cup is available	True / False

Coffee Shop Data

<i>Name Field</i>	<i>Type</i>	<i>Meaning</i>	<i>Example</i>
Coffee Shop ID	Hexadecimal	The unique identifier for each participating coffee shop	0x1234
Coffee Shop Name	String	The name for each participating coffee shop	Esc Cafe
Loyalty Program	String	The name of the Coffee Shops setup loyalty program	Esc Cafe Loyalty Program

System Data

<i>Name Field</i>	<i>Type</i>	<i>Meaning</i>	<i>Example</i>
Admin Account Name	String	Account Name for admin access to the system	AdminUser
Admin Account Password	String	Account password for admin access to the system	AdminPassword123
Secondary Authentication	Integer	Secondary Authentication for admin access into the system (i.e. 2nd factor auth through phone)	123 456 789
System Status	Boolean	Current system status, inputted true would display as Online and false would display as Offline	Online or Offline

Analysis and Design

Class Diagram

A class diagram is a basic blueprint for creating software systems. Its value goes beyond basic visualisation; it is critical in system development and comprehen-

sion. A class diagram provides a thorough picture of the software’s architecture by displaying the structure of a system through classes, characteristics, methods, and their interactions. It serves as a common language for developers and stakeholders, ensuring that all system components and interactions are understood. This common knowledge aids in better communication, resulting in fewer misunderstandings and smoother collaboration among team members. Class diagrams also assist developers make educated judgements about how classes should interact, which properties they should have, and which methods they should expose.

Classes Coffee Cup Subscription System: The central controller of the entire system, arranging interactions between classes. It regulates subscription management, user control, and interconnections with coffee shops, collection points, and the mobile app. Its importance lies in maintaining system integrity and ensuring smooth operations (Djedjiga Mouheb, 2016).

User: A fundamental class representing subscribers to the coffee cup service. Users have distinct attributes such as Name, Contact Information, and Subscription Details. They access methods like `PurchaseSubscription()` for service initiation and `ViewSubscriptionStatus()` to monitor subscription progress. Users are central to the system, as they drive the subscription process and serve as the primary stakeholders.

Coffee Cup: Represents individual reusable coffee cups, each with a unique Tracking ID for tracking and management purposes. While this class doesn’t contain specific methods in the diagram, it plays a vital role in tracking cup usage and returning cups to the system.

Coffee Shop: Depicts participating coffee shops. It maintains information about each shop’s Location. While it doesn’t have many specific methods in this diagram, a `CoffeeShop` class could include actions related to coffee shop participation and cup management.

RFID Tag: The RFID Tag associated with each Coffee Cup serves as a unique identifier. This tag allows the system to track and manage each reusable coffee cup individually. Although the diagram doesn’t specify methods for this class, the RFID Tag is crucial for monitoring the usage of the cups and ensuring they are returned to the system properly.

Collection Point: Represents designated cup collection points, often located strategically for user convenience. It shares attributes like Location. Similar to `CoffeeShop`, it may include methods to control collection points and manage cup returns.

MobileApp: The mobile application that allows users to manage subscriptions and access the system. This class, while not displaying specific attributes in the diagram, might include user-specific data. Methods such as `ManageSubscription()` and `ViewSubscription()` offer user control over their subscriptions, making the

app a crucial user interface.

Protection: Manages safety features, especially in emergency situations. It doesn't have specific methods in the diagram, but it plays a critical role in guaranteeing user safety during unexpected incidents. It can trigger emergency protocols and coordinate safety measures. It also includes the safety of user data and other sensitive information through use of encryption and further privacy systems.

Server: Facilitates various aspects of system control and customization based on user preferences. Methods like provide users with the ability to personalise their system experience, contributing to enhanced user satisfaction and system adaptability (Barclay and Savage, 2003).

Figure 2.0 Class Diagram

State Diagram

A state diagram is a graphical depiction of the numerous states that an item, system, or process might be in and how they change in response to events or situations. It is critical for modelling and comprehending complex systems, particularly in software and control engineering (Barclay, 2003). State diagrams represent a system's behaviour in a clear and understandable manner, making it easier to recognise and analyse how it responds to various events. They are useful in system design, ensuring that systems operate correctly and reliably by capturing the flow of states and transitions, making them a critical tool in software and hardware design and control systems.

Figure 3.1 | State Diagram - Coffee Cup Subscription *Diagram Description* The state diagram illustrates the various states and transitions related to the management of subscription status within a coffee cup service. It depicts how users can interact with the service and make decisions regarding their subscription.

States and Transitions:

Not Subscribed (Initial State):

This is the initial state representing users who have not yet subscribed to the coffee cup service. Users in this state have not activated a subscription. Transition to "Subscribed" occurs when a user decides to subscribe.

Subscribed:

This state represents users who have successfully subscribed to the coffee cup service. Users in this state have an active subscription and can enjoy the benefits of the service, such as purchasing reusable cups. Transition to "Not Subscribed" occurs when a user decides to cancel their subscription. Transition to "On Hold" occurs when a user chooses to temporarily pause their subscription.

On Hold:

This state represents users who have temporarily paused their subscription. Users in this state may not actively use the service, but they have the option to resume their subscription. Transition to “Subscribed” occurs when a user decides to resume their subscription. Transition to “Not Subscribed” occurs when a user decides to cancel their subscription while it’s on hold.

User Interactions:

Subscription Activation (Not Subscribed to Subscribed): Users who are not subscribed can activate a subscription, which transitions them to the “Subscribed” state.

Subscription Cancellation (Subscribed to Not Subscribed): Users with an active subscription can cancel it, reverting to the “Not Subscribed” state.

Subscription Pausing (Subscribed to On Hold): Users with an active subscription can choose to pause it temporarily, moving to the “On Hold” state.

Subscription Resumption (On Hold to Subscribed): Users with a paused subscription can decide to resume it, returning to the “Subscribed” state.

Cancellation During Pause (On Hold to Not Subscribed): Users with a paused subscription can also choose to cancel it, transitioning back to the “Not Subscribed” state.

Additional Notes:

The diagram represents the flexibility and options available to users regarding their coffee cup subscription status. It helps in understanding how users can transition between different states based on their choices. This diagram serves as a visual representation of the subscription management system within the coffee cup service, facilitating system design and development.

Figure 3.2 | State Diagram - Resuable Coffee Cup Lifecycle *Diagram Description:*

This state represents the stages of the reusable coffee cup during its lifecycle throughout the system.

States and Transitions:

Purchased: When a customer purchases the reusable coffee cup.

In Use: When the customer is using the cup.

Returned: When the customer returns the cup to a collection point.

Cleaned: After the cup is cleaned at the collection point.

Available: When the cup is available for another customer to purchase.

Transitions:

From Purchased to In Use when the customer starts using the cup.

From In Use to Returned when the customer returns the cup.

From Returned to Cleaned after the cup is cleaned.

From Cleaned to Available when the cup is ready for reuse.

From Available to Purchased when another customer purchases the cup.

Requirements Traceability Matrix

<i>Requirement ID</i>	<i>Use Cases</i>	<i>Classes</i>	<i>Methods</i>	<i>Packages</i>
Functional 1	Use Cases 1, 2, 3	RFID Tag, Collection Points	Track Collection	App
Functional 2	Use Case 2	Users	Status	App
Functional 3	Use Case 3	Users	Login, Account Creation	App
Functional 4	Use 1, 3	Users	Purchasing	App
Functional 5	Use Case 1, 2, 3	Users, Collection Points	Payment	App
Non-Functional 1	N/A	N/A	N/A	N/A
Non-Functional 2	N/A	Users	Encryption	App
Non-Functional 3	N/A	N/A	N/A	N/A
Non-Functional 4	N/A	Users, Collection Points	Encryption	App

List of design assumptions (if any)

This will help the reader to understand why you have done certain things. Please review the assumptions carefully before submission. (But note: A poor assumption should not be used as an excuse for poor design decisions.)

Test specifications

Functional Requirement 1: RFIP Tracking

Test ID	TCF01.1
Test Name	RFIP Tag Tracking
Test Description	Verify that the RFIP tags can track the cup's collection points, including where it was purchase from.

Requirements Tested	Functional Requirement 1
Input Specification	RFID-tagged cups, RFIP readers / scanners, Tracking Software.
Testing Procedure	1. Attach an RFID-tagged cup to an RFID reader/scanner. 2. Initiate tracking using the tracking software. 3. Record the collection point and purchase location information.
Output Specifications	The RFID system should accurately track the cup's collection points and purchase locations.

Test ID	TCF01.2
Test Name	Multiple RFID Tag Tracking
Test Description	Verify that the RFID system can track multiple cups simultaneously.
Requirements Tested	Functional Requirement 1
Input Specification	Multiple RFID-tagged cups, RFID readers/scanners, tracking software.
Testing Procedure	1. Attach multiple RFID-tagged cups to RFID readers/scanners. 2. Simultaneously initiate tracking for all cups using the tracking software. 3. Verify that the system accurately tracks all cups and their collection points.
Output Specifications	The RFID system should accurately track multiple cups and their collection points.

Test ID	TCF01.3
Test Name	Real-time RFID Tracking
Test Description	Verify that the RFID tracking system provides real-time tracking information.
Requirements Tested	Functional Requirement 1
Input Specification	RFID-tagged cups, RFID readers/scanners, tracking software, real-time tracking display.

Testing Procedure	1. Attach an RFID-tagged cup to an RFID reader/scanner. 2. Initiate tracking using the tracking software. 3. Verify that real-time tracking information is displayed.
Output Specifications	The RFID system should provide real-time tracking information for the cup's collection points.

Test Plan | Functional Requirement 1

<i>Test Schedule</i>	<i>Start Date:</i> Day 1 <i>End Date:</i> Day 10
<i>Testing Resources Required</i>	RFID-tagged cups, RFID readers/scanners, Tracking software, Test environment
<i>Testing Milestones:</i>	<i>Milestone 1: Day 3:</i> This milestone will focus on testing the basic functionality of RFID tag tracking. Test cases will include attaching RFID-tagged cups to RFID readers/scanners and ensuring that the tracking system accurately records collection points and purchase locations. <i>Milestone 2: Day 6:</i> In this milestone, the system's ability to track multiple cups simultaneously will be tested. Multiple RFID-tagged cups will be attached to RFID readers/scanners, and the system's performance in tracking all cups will be assessed. <i>Milestone 3: Day 10:</i> The final milestone will evaluate the real-time tracking capabilities of the RFID system. Users will attach RFID-tagged cups, initiate tracking using the software, and verify that real-time tracking information is provided.
<i>Testing Deliverables</i>	Test results and logs for each milestone

Functional Requirement 2: User Indication

Test ID	TCF02.1
Test Name	User Cup Purchase Indication
Test Description	The app must indicate whether or not the user has purchased a reusable cup.
Requirements Tested	Functional Requirement 2
Input Specification	Mobile devices with the app installed.
Testing Procedure	1. Open the mobile app. 2. Access the user's profile. 3. Verify the presence of an indication regarding the purchase of a reusable cup.
Output Specifications	The app should indicate whether the user has purchased a reusable cup.

Test ID	TCF02.2
Test Name	Non-Purchase User Indication
Test Description	Verify that the app indicates when the user has not purchased a reusable cup.
Requirements Tested	Functional Requirement 2
Input Specification	Mobile app, user profile access.
Testing Procedure	1. Open the mobile app 2. Access the user's profile. 3. Verify the presence of an indication regarding the non-purchase of a reusable cup.
Output Specifications	The app should indicate when the user has not purchased a reusable cup.

Test ID	TCF02.3
Test Name	Indication Consistency
Test Description	Confirm the consistency of user indications in different app sections.
Requirements Tested	Functional Requirement 2
Input Specification	Mobile app, app section access.
Testing Procedure	1. Open the mobile app. 2. Access different sections within the app (e.g., home, profile). 3. Verify the consistency of user indications regarding reusable cup purchase.

Output Specifications	User indications should remain consistent across different app sections.
------------------------------	--

Test Plan | Functional Requirement 2

Test Schedule	<i>Start Date:</i> Day 11 <i>End Date:</i> Day 20
Testing Resources Required	Mobile devices with the app installed, Test accounts
Testing Milestones:	<i>Milestone 1: Day 13:</i> This milestone will focus on verifying that the app can accurately indicate whether A user has purchased a reusable cup. Users will access their profiles and verify the presence of purchase indications. <i>Milestone 2: Day 16:</i> In this milestone, the absence of a reusable cup purchase will be tested. Users will access their profiles and verify the presence of indications indicating the non-purchase of a reusable cup. <i>Milestone 3: Day 20:</i> The final milestone will focus on ensuring the consistency of user indications across different sections of the app. Users will access different app sections (e.g., home, profile) and verify that indications remain consistent.
Testing Deliverables	Test results and logs for each milestone

Functional Requirement 3: User Authentication

Test ID	TCF03.1
Test Name	User Login Functionality
Test Description	Verify that the app's login function allows users to sign in and create personal accounts.
Requirements Tested	Functional Requirement 3

Input Specification	Mobile app, login process, user credentials.
Testing Procedure	1. Open the mobile app. 2. Initiate the login process. 3. Enter user credentials or create a new account. 4. Verify successful login or account creation.
Output Specifications	The app should allow users to sign in and create personal accounts.

Test ID	TCF03.2
Test Name	User Authentication Consistency
Test Description	Confirm the consistency of user authentication processes.
Requirements Tested	Functional Requirement 3
Input Specification	Mobile app, login process, user credentials.
Testing Procedure	1. Open the mobile app. 2. Initiate the login process. 3. Enter user credentials or create a new account. 4. Verify that the authentication process remains consistent over multiple attempts.
Output Specifications	The app's authentication process should remain consistent.

Test ID	TCF03.3
Test Name	Invalid Login Handling
Test Description	Verify that the app handles invalid login attempts appropriately.
Requirements Tested	Functional Requirement 3
Input Specification	Mobile app, login process, incorrect user credentials.
Testing Procedure	1. Open the mobile app. 2. Initiate the login process. 3. Enter incorrect user credentials. 4. Verify that the app handles invalid login attempts as specified.
Output Specifications	The app should handle invalid login attempts appropriately.

Test Plan | Functional Requirement 3

Test Schedule	<i>Start Date:</i> Day 21 <i>End Date:</i> Day 30
Testing Resources Required	Mobile devices with the app installed, Test accounts
Testing Milestones:	<i>Milestone 1: Day 23:</i> This milestone will focus on verifying that the app's login function allows users to sign in and create personal accounts. Users will initiate the login process, enter user credentials or create new accounts, and verify successful login or account creation. <i>Milestone 2: Day 26:</i> This milestone will ensure the consistency of user authentication processes. Users will initiate the login process multiple times and verify that the authentication process remains consistent. <i>Milestone 3: Day 30:</i> The final milestone will focus on verifying that the app handles invalid login attempts appropriately. Users will initiate login with incorrect credentials and verify that the app handles invalid login attempts as specified.
Testing Deliverables	Test results and logs for each milestone

Functional Requirement 4: User Purchase Options

Test ID	TCF04.1
Test Name	Cup and Coffee Purchase
Test Description	Verify that the app supports the purchase of both coffee and reusable cups.
Requirements Tested	Functional Requirement 4
Input Specification	Mobile app, purchase section, payment methods.

Testing Procedure	1. Open the mobile app. 2. Navigate to the purchase section. 3. Attempt to purchase both coffee and a reusable cup. 4. Use different payment methods for testing.
Output Specifications	The app should support the purchase of both coffee and reusable cups with various payment methods
Test ID	TCF04.2
Test Name	Payment Method Availability
Test Description	Verify that the app offers various payment methods for user selection.
Requirements Tested	Functional Requirement 4
Input Specification	Mobile app, payment options.
Testing Procedure	1. Open the mobile app. 2. Navigate to the payment options during a purchase. 3. Verify the availability of credit cards, debit cards, mobile wallets, and PayPal as payment methods.
Output Specifications	The app should offer the specified payment methods for user selection.
Test ID	TCF04.3
Test Name	Payment Process Validation
Test Description	Verify the accuracy of the payment process.
Requirements Tested	Functional Requirement 4
Input Specification	Mobile app, payment section, payment methods.
Testing Procedure	1. Open the mobile app. 2. Navigate to the payment section. 3. Initiate a payment using different methods. 4. Verify the accuracy of the payment process, including transaction confirmation.
Output Specifications	The app should accurately process payments using the specified payment methods

Test Plan | Functional Requirement 4

Test Schedule	<i>Start Date:</i> Day 31 <i>End Date:</i> Day 40
Testing Resources Required	Mobile devices with the app installed, Test accounts, Payment methods for testing
Testing Milestones:	<i>Milestone 1: Day 33:</i> This milestone will focus on verifying that the app supports the purchase of both coffee and reusable cups. Users will navigate to the purchase section, attempt to purchase both coffee and a reusable cup, and use different payment methods for testing. <i>Milestone 2: Day 36:</i> In this milestone, the availability of various payment methods will be tested. Users will navigate to the payment options during a purchase and verify the availability of credit cards, debit cards, mobile wallets, and PayPal as payment methods. <i>Milestone 3: Day 40:</i> The final milestone will focus on verifying the accuracy of the payment process. Users will navigate to the payment section, initiate payments using different methods, and verify the accuracy of the payment process, including transaction confirmation.
Testing Deliverables	Test results and logs for each milestone

Functional Requirement 5: Privacy Policy Availability

Test ID	TCF05.1
Test Name	Privacy Policy Access
Test Description	Verify that the service provides access to its privacy policy.
Requirements Tested	Functional Requirement 5
Input Specification	Service access, internet connection, policy documentation.

Testing Procedure	1. Access the service (website or app) 2. Navigate to the policy section. 3. Verify the availability and accessibility of the privacy policy.
Output Specifications	The service should provide access to its privacy policy.

Test ID	TCF05.2
Test Name	Privacy Policy Content
Test Description	Verify that the privacy policy contains relevant and comprehensive information.
Requirements Tested	Functional Requirement 5
Input Specification	Service access, internet connection, policy documentation.
Testing Procedure	1. Access the service (website or app). 2. Navigate to the privacy policy section. 3. Review the content of the privacy policy. 4. Verify that it contains relevant and comprehensive information about data collection and usage.
Output Specifications	The privacy policy should contain relevant and comprehensive information.

Test ID	TCF05.3
Test Name	Terms of Use Access
Test Description	Verify that the service provides access to its terms of use.
Requirements Tested	Functional Requirement 5
Input Specification	Service access, internet connection, policy documentation.
Testing Procedure	1. Access the service (website or app). 2. Navigate to the terms of use section. 3. Verify the availability and accessibility of the terms of use.
Output Specifications	The service should provide access to its terms of use.

Test Plan | Functional Requirement 5

Test Schedule	<i>Start Date:</i> Day 41 <i>End Date:</i> Day 50
Testing Resources Required	Access to the service (website or app), Internet connection, Policy documentation
Testing Milestones:	<i>Milestone 1: Day 43:</i> This milestone will focus on verifying that the service provides access to its privacy policy. Users will access the service (website or app), navigate to the policy section, and verify the availability and accessibility of the privacy policy. <i>Milestone 2: Day 46:</i> In this milestone, the content of the privacy policy will be reviewed. Users will access the service, navigate to the privacy policy section, and review the content to ensure it contains relevant and comprehensive information about data collection and usage. <i>Milestone 3: Day 50:</i> The final milestone will focus on verifying that the service provides access to its terms of use.
Testing Deliverables	Test results and logs for each milestone

Non-Functional Requirement 1: Response Time

Test ID	TCNF01.1
Test Name	Login Page Response Time.
Test Description	Verify that the login page grants users access to the system within 3 seconds.
Requirements Tested	Non Functional Requirement 1
Input Specification	Login initiation.
Testing Procedure	1. Open the mobile app. 2. Initiate the login process. 3. Measure the time taken for the page to load.
Output Specifications	The login page should load within 3 seconds.

Test ID	TCNF01.2
----------------	----------

Test Name	Network Variability Testing.
Test Description	Test the login page's response time under various network conditions.
Requirements Tested	Non Functional Requirement 1
Input Specification	Login initiation, network simulation
Testing Procedure	1. Open the mobile app. 2. Initiate the login process under different network conditions (e.g., slow, unstable). 3. Measure the time taken for the page to load in each scenario.
Output Specifications	The login page should meet response time requirements under various network conditions.

Test Plan | Non-Functional Requirement 1

<i>Test Schedule</i>	<i>Start Date:</i> Day 1 <i>End Date:</i> Day 5
<i>Testing Resources Required</i>	Mobile devices with the app installed, Test accounts
<i>Testing Milestones:</i>	<i>Milestone 1: Day 3:</i> This milestone will focus on verifying that the login page grants users access to the system within 3 seconds. Users will access the login page and measure the response time. <i>Milestone 2: Day 5:</i> In this milestone, the overall response time of the application will be tested. Users will navigate through the app and measure the response time for various actions.
<i>Testing Deliverables</i>	Test results and response time measurements for each milestone.

Non-Functional Requirement 2: Compliance

Test ID	TCNF02.1
Test Name	GDPR Compliance.
Test Description	Verify that the system complies with GDPR data protection laws.
Requirements Tested	Non Functional Requirement 2
Input Specification	Compliance documentation, GDPR standards.

Testing Procedure	1. Review the system's compliance documentation. 2. Verify that the system adheres to GDPR requirements.
Output Specifications	The system should comply with GDPR data protection laws.

Test Plan | Non-Functional Requirement 2

Test Schedule	<i>Start Date:</i> Day 6 <i>End Date:</i> Day 10
Testing Resources Required	Compliance documentation, Test environment
Testing Milestones:	<i>Milestone 1: Day 8:</i> This milestone will focus on verifying the system's compliance with GDPR. Compliance documentation will be reviewed to ensure adherence to GDPR standards. <i>Milestone 2: Day 10:</i> In this milestone, the system's compliance with environmental standards will be tested. Compliance documentation and environmental impact reports will be reviewed.
Testing Deliverables	Compliance assessment reports for each milestone.

Non-Functional Requirement 3: Hardware Compatability

Test ID	TCNF03.1
Test Name	Compatibility with iOS.
Test Description	Verify that the system is compatible with iOS devices.
Requirements Tested	Non Functional Requirement 3
Input Specification	iOS devices, app installation.
Testing Procedure	1. Open the mobile app on iOS devices. 2. Verify functionality and compatibility.
Output Specifications	The system should be compatible with iOS devices.

Test ID	TCNF03.2
----------------	----------

Test Name	Compatibility with Android.
Test Description	Verify that the system is compatible with Android devices.
Requirements Tested	Non Functional Requirement 3
Input Specification	Android devices, app installation.
Testing Procedure	1. Open the mobile app on Android devices. 2. Verify functionality and compatibility.
Output Specifications	The system should be compatible with Android devices.

Test Plan | Non-Functional Requirement 3

<i>Test Schedule</i>	<i>Start Date:</i> Day 11 <i>End Date:</i> Day 15
<i>Testing Resources Required</i>	Test devices (IOS and Android), Test accounts
<i>Testing Milestones:</i>	<p><i>Milestone 1: Day 13:</i> This milestone will focus on verifying the system's compatibility with IOS devices. Testing will be conducted using IOS devices to ensure proper functionality.</p> <p><i>Milestone 2: Day 15:</i> In this milestone, the system's compatibility with Android devices will be tested. Testing will be conducted using Android devices to ensure proper functionality.</p>
<i>Testing Deliverables</i>	Compatibility test results and device compatibility reports for each milestone.

Non-Functional Requirement 4: Authentication and Encryption

Test ID	TCNF04.1
Test Name	Data Encryption Validation
Test Description	Verify that data encryption is available to protect personal data.
Requirements Tested	Non Functional Requirement 4
Input Specification	User account access, personal data.

Testing Procedure	1. Open the mobile app. 2. Access personal data. 3. Verify the presence of data encryption.
Output Specifications	Data encryption should be available to protect personal data.

Test ID	TCNF04.2
Test Name	Authentication Process Validation
Test Description	Test the app's authentication process for user data protection.
Requirements Tested	Non Functional Requirement 4
Input Specification	User account access, authentication process.
Testing Procedure	1. Open the mobile app. 2. Initiate the authentication process. 3. Enter user credentials. 4. Verify successful authentication.
Output Specifications	The app's authentication process should protect personal data as specified.

Test ID	TCNF04.3
Test Name	Authentication Process Consistency
Test Description	Confirm the consistency of the authentication process.
Requirements Tested	Non Functional Requirement 4
Input Specification	User account access, authentication process.
Testing Procedure	1. Open the mobile app. 2. Initiate the authentication process multiple times. 3. Enter user credentials. 4. Verify that the authentication process remains consistent over multiple attempts.
Output Specifications	The app's authentication process should consistently protect personal data.

TestPlan | Non-Functional Requirement 4

Test Schedule	<i>Start Date:</i> Day 16 <i>End Date:</i> Day 20
Testing Resources Required	Mobile devices with the app installed, Test accounts
Testing Milestones:	<i>Milestone 1: Day 18:</i> This milestone will focus on verifying the availability and effectiveness of data encryption in the system. Testing will involve sending and receiving encrypted data. <i>Milestone 2: Day 20:</i> In this milestone, user authentication processes will be tested for security. Testing will include login attempts and authentication verification.
Testing Deliverables	Security test results and authentication reports for each milestone

Non-Functional Requirement 5: Policy Compliance

Test ID	TCNF05.1
Test Name	Privacy Policy Availability
Test Description	Verify that the service provides a privacy policy.
Requirements Tested	Non Functional Requirement 5
Input Specification	Service access, policy documentation.
Testing Procedure	1. Access the service, website, or app. 2. Verify the availability and accessibility of the privacy policy.
Output Specifications	The service should provide a privacy policy for users.

Test ID	TCNF05.2
Test Name	Terms of Use Availability
Test Description	Verify that the service provides terms of use.
Requirements Tested	Non Functional Requirement 5
Input Specification	Service access, policy documentation.
Testing Procedure	1. Access the service, website, or app. 2. Verify the availability and accessibility of the terms of use.

Output Specifications	The service should provide terms of use for users.
Test ID	TCNF05.3
Test Name	Data Collection Transparency
Test Description	Verify that the service clearly states its data collection processes.
Requirements Tested	Non Functional Requirement 5
Input Specification	Service access, policy documentation.
Testing Procedure	1. Access the service, website, or app. 2. Review the privacy policy or terms of use. 3. Verify that data collection processes are transparently explained.
Output Specifications	The service should clearly state its data collection processes in the provided policies.

TestPlan | Non-Functional Requirement 5

<i>Test Schedule</i>	<i>Start Date:</i> Day 21 <i>End Date:</i> Day 25
<i>Testing Resources Required</i>	Access to the service (website or app), Internet connection, Policy documentation
<i>Testing Milestones:</i>	<i>Milestone 1: Day 23:</i> This milestone will focus on verifying that the service provides access to its privacy policy. Users will access the service, navigate to the policy section, and verify the availability and accessibility of the privacy policy. <i>Milestone 2: Day 25:</i> The final milestone will focus on verifying that the service provides access to its terms of use.
<i>Testing Deliverables</i>	Compliance verification reports for each milestone

Project Management

Minimal Viable Product

The minimal viable product will be a basic version of the CCSMS that includes the necessary requirements. These requirements implemented must include all

the functional requirements and does not have to include the non-functional ones. These required requirements include the use of RFID tags, if users have purchased coffee cups, a login feature, the option to purchase coffee and reusable cups, the use of payment methods through the app. It will also need all the components necessary for the architecture. These will include the application, online servers for tracking and accounts, the coffee cups and at least 1 coffee shop participating.

Milestones

The milestones listed below are part of the implementation phase of our project that also lead to our final product. It is essential everyone working on the project stays on track with them so that the final product can be released successfully.

Milestone 1: Review of the requirements

Description: The requirements are reviewed to ensure that they will definitely meet the customer's needs and expectations.

Milestone 2: Software development for MVP

Description: Software development is started on the MVP and ensures it is meeting the basic requirements.

Milestone 3: Creation of the MVP

Description: The MVP is created once the software development is completed. This will include the functions as well as the interfaces for all the applications and servers.

Milestone 4: Testing of MVP

Description: Real test users should be able to use our MVP with ease. If the reviews from the people testing the MVP are satisfactory, this milestone can be completed.

Milestone 5: Development of final product

Description: Development of the final product can begin and will include the implementation of the rest of the requirements.

Milestone 6: Release of final product

Description: Final product is released and users can give feedback on if it meets their desires.

Milestone 7: Maintenance of final product

Description: Ongoing maintenance of the final product is done to ensure any bugs or issues are fixed and to make any other improvements.

Tasks

<i>Task ID</i>	<i>Description</i>	<i>Dependencies</i>	<i>Effort (S-XL)</i>	<i>Milestone</i>
1	Review requirements necessary for the MVP	None	M	1
2	Application and servers are setup	None	L	2
3	The basic code is written for the MVP	2	L	2
4	Test coffee cups are collected and connected to the application	2, 3	L	2
5	The MVP is released	1, 2, 3, 4	XL	3
6	Users test how the features work on the MVP. If the tests are satisfactory, this task can be completed.	5	M	4
7	Non-functional requirements and other features are added to lead to the final product	6	L	5
8	Product is launched	7	XL	6
9	User feedback is reviewed	8	L	7

<i>Task ID</i>	<i>Description</i>	<i>Dependencies</i>	<i>Effort (S-XL)</i>	<i>Milestone</i>
10	Bugs and improvements are added based on feedback	9	XL	7

Risks

<i>Risk ID</i>	<i>Description</i>	<i>Probability</i>	<i>Severity</i>	<i>Issues Caused</i>	<i>Mitigation Strategies</i>
1	Change in stakeholders	30%	70%	Project is slowed down and new stakeholders have to be found.	Find reliable stakeholders and ensure they are in favour of the product.
2	Team member is not available to work on project anymore	5%	80%	Team members have to be replaced or more work has to be done by others.	Ensure members have the necessary qualifications and are responsible. Some aspects of this risk are unavoidable (e.g. death or injury).
3	Change in functional requirements	20%	70%	Features may have to be added or removed from the MVP.	Ensure functional requirements are not misleading or not thought out well.
4	Change in non-functional requirements	30%	50%	Features may have to be added or removed from the final product.	Ensure non-functional requirements are written out well and the team is confident they can manage them.

<i>Risk ID</i>	<i>Description</i>	<i>Probability</i>	<i>Severity</i>	<i>Issues Caused</i>	<i>Mitigation Strategies</i>
5	Code is low quality	5%	90%	Software design will have to be changed.	Ensure all software engineers and programmers on the project are very confident and have qualifications necessary for writing code.
6	Software is not up to date	8%	80%	Software used will have to be changed.	Ensure software used is consistently still updated and is not an old version of the software.
7	Programming language choice was poor	15%	80%	Coding language will have to be changed.	Ensure the language chosen will be appropriate for the CCSMS project.
8	Servers are poor quality	15%	85%	Servers will have to be changed.	Ensure servers chosen are reliable and easy to connect to.

Summary and Outlook

The conceptual development of the Coffee Cup Subscription Management System has been an ambitious project for the whole team and our stakeholders. This project aimed to create a in-depth solution to streamline sustainable coffee culture, changing the way coffee cups are used, managed, and tracked. With a three-layered architecture alongside a complex data strategy proposition, and in-depth analysis, we were able to define and supply our stakeholders with a viable concept for our project, defining our vision for this journey.

Rigorous testing of all aspects of the CCSMS project were thoughtout, with a multitude of tests selected to supply a guaranteed product. Milestones were planned to ensure that the project flow will maintain throughout development, understanding where the system should be at and what achievements should have already been obtained and any moment along the project development

lifespan.

In all, the CCSMS project capitalised off the power of effective teamwork, adaptability and brainstorming abilities to build a solid foundation for this document. The outlook for CCSMS is promising and is ready to make a meaningful impact in the world of coffee cup consumption and sustainable practices.

Appendices

Condensed Log with Stakeholders

Date	Attended	Apologies	Summary
25/09/23	All	None	Initial SRS Handover Report Discussion - How we use the document as a basis to build off
02/10/23	All	None	Discussed how we would meet all the functional requirements and discussed necessity of each non-functional requirements
09/10/23	All	None	Discussed how we would break down each task and get it done on schedule
16/10/23	All-2	Md Tanvir, Blake	Outlined testing procedures with stakeholders - finalised on 3 tests for each functional requirement and 1-3 tests for each decided non-functional requirement

Date	Attended	Apologies	Summary
23/10/23	All	None	Updated stakeholders on current situation and where we are in schedule
30/10/23	All	None	Presented Overview on the SDD Document for stakeholders

Third-party-resources

Third-party Resource	Authors	Location	Access-date
Lucid Chart	Lucid Team	https://lucid.co	23/10/23
Draw.io	Draw.io Team	drawio.com	26/10/23
Mermaid	Mermaid Team	mermaid.js.org	26/10/23
“(2003). Object-Oriented Design with UML and Java. Elsevier.”	Barclay, K. and Savage, J.	Page 15-20	28/10/23
“Aspect-oriented security hardening of uml design models.”	Djedjiga Mouheb	https://www.researchgate.net/publication/275521215_Aspect-Oriented_Security_Hardening_of_UML_Design_Models	28/10/23
“Understanding UML : the developer’s guide : with a Web-based application in Java.”	Harmon, P. and Watson, M.	Page 240-250	28/10/23