Software Requirements Specification (SRS) Document

Team Number : 10

Members:

Gowlapalli Rohit

Losetti Mourya

Vempati Siva Koti Reddy

Kadali Lakshmi Nirmala

Brief problem statement

The Airdrop Insurance initiative in the Gurue District of Mozambique aims to provide a social safety net for the financially disadvantaged through a voucher system. This system can be accessed through a mobile platform, allowing businesses & residents to purchase personal and crop insurance. The vouchers can be redeemed for the local digital currency or other goods. The program also leverages collective risk to gain stronger bargaining power in reinsurance negotiations. Our aim is to improvise the app that has been provided and to remove various privacy issues from it.Our Primary Goal is to resolve privacy issues and deploy it in google play store.

Users profile

- **Members:** Users who apply for insurance using digital tokens. They may not be familiar with computer or software usage. Eg: Farmers, Jobholders, etc.
- **Community Manager :** Verifies official documents and audio recordings to grant User the access to the Airdrop Application.
- **Admin:** Administrates the Airdrop Application.
- **Speculator:** Bids & cashes in the members' insurances and takes on the responsibility of managing payouts to the users while also assuming the risk involved.

Project Modules

- **User Management Module**: The User Management module enables administrators to manage user accounts, permissions, and access to app features. It provides features such as user authentication, registration, password reset, and user profile management. The module is integrated with Firebase Authentication, which enables secure and scalable user authentication and management.
- **Insurance Management Module :** It is the process of carrying out and putting into action the use of reinsurance in a particular insurance program or system. It involves incorporating reinsurance as a risk management strategy. It aims to enhance the stability and sustainability of an insurance program.
- **Digital Transaction Module** : Digital tokens can facilitate faster claim processing and payment. This can help people get the support they need more quickly, which can be critical in times of crisis.
- **Privacy Protection Module:** By obtaining only the essential permissions from the user, we can ensure compliance with Google Play Store's terms and conditions, preventing potential issues during app deployment. This approach helps to avoid any violations of the store's policies.

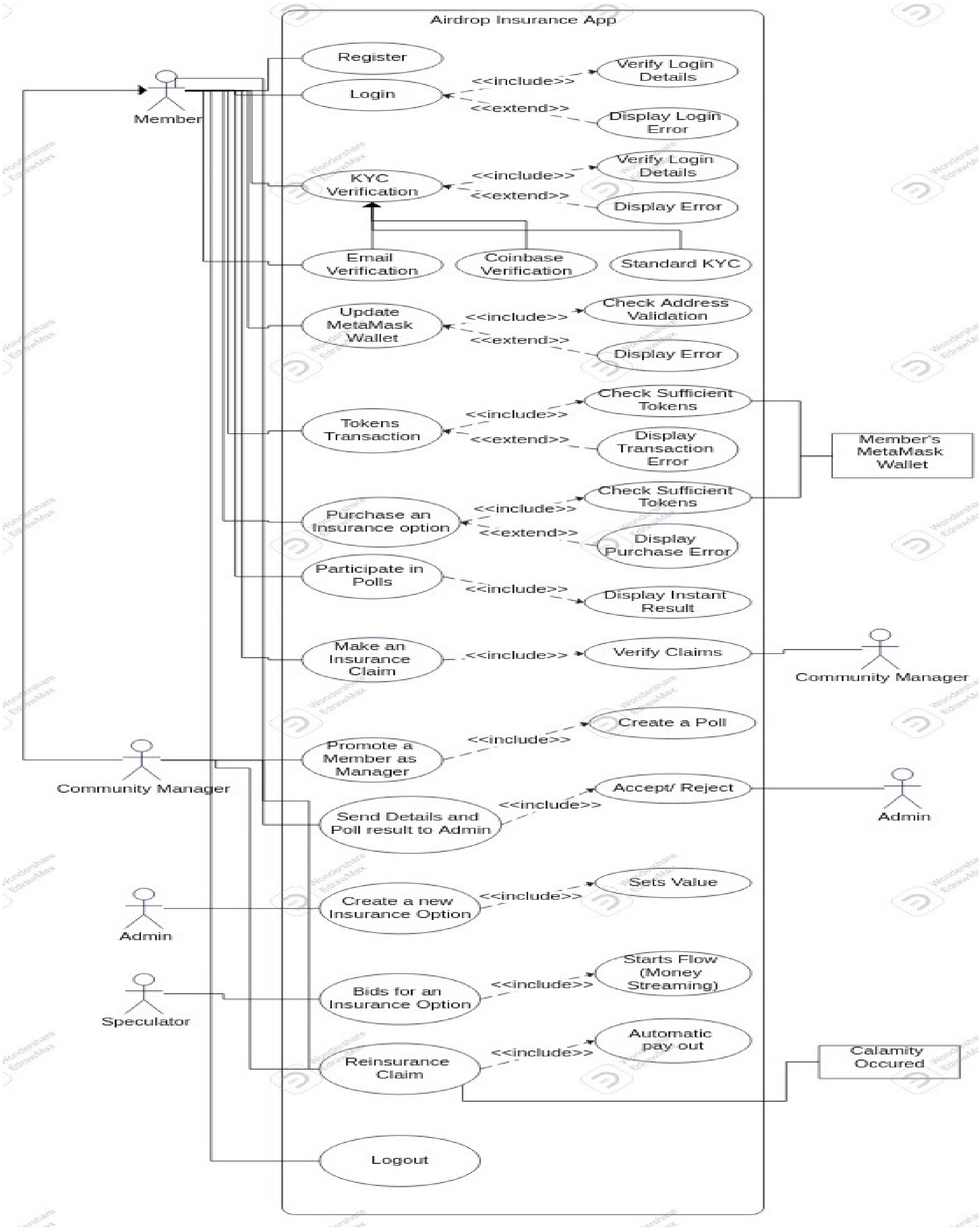
- **Deployment Module:** The deployment module automates the process of packaging, distributing, and installing the app on the Google Play Store and IOS App Store. It is integrated with Flutter and Firebase and provides versioning, testing, monitoring, and app store listing features.
- **Speculator Module** : Creating a new role for transferring risk of a community of members with a specific Insurance Option to a speculator . A speculator bids & cashes in the members' insurances and takes on the responsibility of managing payouts to the users while also assuming the risk involved. We are integrating this role into our app
- **UI Interface Module** : UI/UX interface of a Flutter app would need to be designed to provide a smooth and intuitive experience for users who want to purchase insurance coverage for airdrops.

Feature requirements (described using use cases)

S.No	Use Case Name	Description	Release
1	**Dashboard**	Grids to access the other features	R1
2	**Purchasing Insurance Options via Money Streaming in Superfluid**	Used for instantaneous transactions between the treasury wallet and the buyer of an insurance option when purchased, as well as between the treasury wallet and the speculator when they undertake the risk of an insurance option.	R2
3	**Manager Poll**	Community managers can be elected by the other users through voting in polls. Once a sufficient number of votes are cast, a request is sent to the admin for approval to become a community manager.	R1
4	**Bid for Insurance Options by Speculators**	The Application presents speculators with a list of available insurance options, which are linked to members	R2

		paying recurring premiums. Speculators can purchase these options and take on the associated risk, with the Insurance Option Grid providing an overview of the available options and their details..	
5	**Metamask Wallet Authentication**	Metamask Wallet addresses are used to transfer tokens from one wallet to another. Such wallets are connected to our app using this.	R1
6	**Wallet-to-Wallet Transactions using IIITH Tokens**	Digital tokens are exchanged between wallets using their Metamask addresses.	R1
7	**Automated Insurance Claim**	Upon verification by Community Managers, Speculators promptly disburse insurance claims to members at an accelerated flow rate once the weather pattern-based claim criteria are met.	R2

UML Use Case-Diagram:



Features and User-Stories:

- **User Management Module:**

1) **User Login:** This is a basic login page where users enter their details and sign in to the application.
2) **User Registration:** This is a basic registration page where users enter their details and sign up for the application.
3) **User Verification:** This is done using the coinbase account of the user.
4) **Manager Poll:** Community managers can be elected by the other users through voting in polls. Once a sufficient number of votes are cast, a request is sent to the admin for approval to become a community manager.
5) **Dashboard:** Access to all Grids on the Dashboard is only granted to Application members who have completed their KYC Verification or Coinbase account verification.

User Stories:

- ➢ **User Login:** As a registered user of the application, I want to log in to my account by entering my email address and password so that I can access my saved bookmarks, preferences, and other personalized content.

- ➢ **User Registration:** As a new user, I want to be able to register for an account so that I can access the full range of features and services offered by the application.
- ➢ **User Verification:** As a new user who has registered for an account, I want to be able to verify my account through a secure process so that I can ensure my information is protected and begin using the application's features.
- ➢ **ManagerPoll:** As a member of Airdrop Insurance, I nominate and vote for trustworthy and competent community managers to ensure fair distribution. The app sends admin approval request upon sufficient votes for transparency
- ➢ **ManagerPoll:** As an admin of the community platform, I receive a request for approval when sufficient votes are cast for a user to become a community manager. I review the nomination and approve it, ensuring a fair and transparent process for the platform's management.
- ➢ **Dashboard:** As a user of the Airdrop Insurance App, I want to have a dashboard where I can view a summary of my activity, access my recent content and settings, and quickly navigate to other parts of the application.

This will be done in R1.

- **Privacy Protection Module:**

(1) **<u>Resolving Existing Privacy Issues:</u>** The application is requesting access to the External-Storage on the device, even though it only needs permissions for audio, microphone, and some specific documents.

To address this, the **MANAGE_EXTERNAL_STORAGE** permission can be removed from the list of required permissions.

<u>User Stories:</u>

➢ As a user of the Application, I notice it is requesting access to my external storage, which is unnecessary. I am pleased to learn that the app has removed the permission requirement, ensuring my privacy and data security.

This will be done in R1

- **Deployment Module:**

1) **<u>Alpha Testing:</u>** The Application goes through alpha testing after the deployment. The purpose of alpha testing is to identify any major issues or bugs in the software, and to assess the software's overall functionality and usability.

2) **<u>Beta-Testing:</u>** The Application then goes through beta testing. It is a type of software testing that is performed after the alpha testing phase and before the final release of the software to the general public. The purpose of beta testing is to gather

feedback from a wider group of users about the software's functionality, performance, and usability, and to identify any remaining issues or bugs that were not discovered during alpha testing.

User Stories:

➢ As a user of the Application, I am pleased to learn that the application undergoes alpha and beta testing to ensure functionality and usability. Beta testing gathers feedback from a wider group to identify any remaining issues before the final release.

This will be done in R1. (Initial Deployment)

This will be done in R2. (Final Deployment)

- **Digital Transaction module**:
    1) **Wallet-to-Wallet Transactions via SuperTokens** : Sending Tokens to wallet addresses to promote awareness of a new virtual currency . IIITH Tokens are created to test flow of tokens between members using MumbaiPolygon.net

User Stories:

- ➢ As a user of the application, I want to be able to send and receive Supertokens to and from other users' wallets to ease transactional payments.

This will be done in R1

- **Insurance Management module:**
    1) **Purchasing Insurance Options via Money Streaming in Superfluid:** We utilize the Superfluid platform for instantaneous transactions between the treasury wallet and the buyer of an insurance option when purchased, as well as between the treasury wallet and the speculator when they undertake the risk of an insurance option.
    2) **Issuing an Insurance Claim:** Token distribution is automated and is dependent on many factors such as weather conditions, job layoffs etc.

User Stories:

- ➢ **Purchasing Insurance Options via Money Streaming in Superfluid:** As a member of Airdrop Insurance Application, I am able to purchase insurance options through the Superfluid platform, allowing for instantaneous transactions between the treasury wallet and the buyer. The platform also facilitates risk-taking by speculators through the same instantaneous transaction process.

- ➢ **Issuing an Insurance Claim:** As a member of Airdrop Insurance Application, I submit a claim based on predetermined factors such as weather conditions or job layoffs. Token distribution is then automated, ensuring a seamless and efficient process for insurance claims.

This will be done in R2.

- **Speculator Module:**
    1) **Bid for Insurance Options by Speculators:** The Application presents speculators with a list of available insurance options, which are linked to members paying recurring premiums. Speculators can purchase these options and take on the associated risk, with the Insurance Option Grid providing an overview of the available options and their details.

User Stories:

- ➢ As a speculator on the Airdrop Insurance app, I browse the available insurance options linked to recurring premiums and can purchase them to take on the associated risk. The Insurance Option Grid provides an easy-to-use overview of available options and their details.

This will be done in R2

- **UI Interface Module:**
    1) **<u>Improved UI/UX Interface:</u>** Improving the UI/UX design of our app will enhance user engagement and satisfaction, as well as increase retention rates and conversions. An enhanced UI/UX design in the application increases user engagement and satisfaction, resulting in improved retention rates and conversions. The new design incorporates simple symbols to depict mechanisms, eliminating the need for lengthy descriptions of grids in dashboards

<u>User Stories:</u>

➢ As a user of the Airdrop insurance application, I want the UI/UX interface to be more intuitive and user-friendly so that I can navigate the platform more easily and efficiently.

This will be done in R2

Use Case Description

Use Case Number:	**UC-01**
Use Case Name:	**Dashboard**
Overview:	Grids to access the other features
Actors:	Members , Admins ,Speculators and Community-Managers
Pre condition:	Users must have their account registered before logging in.
Flow:	Main (success) flow: 1. The application presents the user with a list of grids to access different features, providing a clear and organized layout for navigation..
	Alternate flows: None
Post condition:	Users should be able to navigate from one page to another that are relevant to them.

Use Case Number:	**UC-02**
Use Case Name:	**Purchasing Insurance Options via Money Streaming in Superfluid**
Overview:	Used for instantaneous transactions between the treasury wallet and the buyer of an insurance option when purchased, as well as between the treasury wallet and the speculator when they undertake the risk of an insurance option.
Actors:	Members of the Application
Pre condition:	To use the application, the user must register in the database and verify their MetaMask wallet.
Flow:	Main (success) flow: 1. The user chooses "InsuranceOptions" from the dashboard menu to purchase a New Insurance Option 2. The user chooses an Insurance Option of choice and clicks on “Purchase”

	button 3. The member purchases an insurance option by transferring digital tokens to the insurance company's treasury wallet on the application platform. 4. The transfer of tokens from the member's wallet to the insurance company's treasury wallet occurs at an agreed-upon flow rate using the money streaming feature on the application. 5. Once the insurance option is purchased, the tokens may be transferred from the insurance company's treasury wallet to a speculator's wallet at a suitable flow rate using the money streaming feature on the application.
	Alternate flows: None
Post condition:	Users should be able to perform token transaction/money streaming using superfluid.

Use Case Number:	**UC-03**
Use Case Name:	**Manager Poll**
Overview:	Community managers can be elected by the other users through voting in polls. Once a sufficient number of votes are cast, a request is sent to the admin for approval to become a community manager.
Actors:	Members of the Application
Pre condition:	The user suggesting the poll must not hold the role of community manager or admin.
Flow:	Main (success) flow: 1. The user chooses "ManagerPoll" from the dashboard menu to create a new poll. 2. To propose a manager poll, the user selects the "+" icon.The user sets the poll start and end dates, proposes the poll, and submits it for approval by the admin.

	3. The user clicks on the “Submit Poll Proposal” button 4. The manager/admin reviews and decides whether to approve the poll or not. 5. If approved, the poll will be displayed in the "ManagerPoll" grid for users to view and participate in.
	Alternate flows: None
Post condition:	A poll will be created in which other users can vote.

Use Case Number:	**UC-04**
Use Case Name:	**Bid for Insurance Options by Speculators**
Overview:	A speculator bids & cashes in the members' insurances and takes on the responsibility of managing payouts to the users while also assuming the risk involved. We are integrating this role into our app.

Actors:	Speculators
Pre condition:	The users must be registered in the database as a speculator.
Flow:	1. The user selects the "Bid Insurance Options" grid from the dashboard. 2. The user selects an insurance option to bid on. 3. The user clicks on the "Purchase" button to complete the bid. 4. If the bid is successful, the user receives a major share of the premiums paid by users for that particular insurance option, but also must pay out if any claims are made by members from their wallet.
	Alternate flows: None
Post condition:	-

Use Case Number:	**UC-05**

Use Case Name:	**Metamask Wallet Authentication**
Overview:	Metamask Wallet addresses are used to transfer tokens from one wallet to another. Such wallets are connected to our app using this.
Actors:	Members , Admins , Speculators and Community-Managers
Pre condition:	Users should be registered in the database.
Flow:	Main (success) flow: 1. The user chooses the "Wallet" option from the dashboard to connect their MetaMask wallet. 2. If the wallet is not yet connected, the user is presented with a button to initiate the connection process.After clicking on the button, the user will be asked to enter his metamask wallet address and metamask private key. 3. The user enters their MetaMask wallet address and private key when prompted.

	4. If the entered details are correct, the user's MetaMask wallet will be connected to the application.
	Alternate flows: If the entered address or key is incorrect, the wallet is not connected.
Post condition:	Using their wallet, the user has the ability to execute token transactions.

Use Case Number:	**UC-06**
Use Case Name:	**Transaction of Tokens using Metamask Wallets**
Overview:	Digital tokens are exchanged between wallets using their Metamask addresses.
Actors:	Members and Community-Managers
Pre condition:	To use the application, the user must register in the database and verify their MetaMask wallet.

Flow:	Main (success) flow: 1. The user selects the "Wallet" option from the dashboard to initiate a token transfer. 2. A form is displayed, asking for the MetaMask wallet address of the recipient and the amount of tokens to be transferred. 3. To complete the token transfer, the user clicks on the icon located to the right of the amount field
	Alternate flows: If the user doesn’t have sufficient balance, a relevant error message is displayed.
Post condition:	The tokens are deducted from the user’s wallet.

Use Case Number:	**UC-07**

Use Case Name:	**Insurance Claim**
Overview:	Upon verification by Community Managers, Speculators promptly disburse insurance claims to members at an accelerated flow rate once the weather pattern-based claim criteria are met.
Actors:	Members of the Application
Pre condition:	The user must have purchased an insurance option beforehand and experienced an unforeseen calamity.
Flow:	Main (success) flow: 1. The user selects the "Insurance Claim" grid from the dashboard. 2. The user is redirected to a view showing their purchased insurance options. 3. The user selects the "Submit Claim" option. 4. The claim is reviewed and, if accepted, the corresponding payout is made by the speculator to the user
	Alternate flows: None

Post condition:	Payout is made by the speculator to the user

Link to UML Case Diagram

https://www.edrawmax.com/online/share.html?code=28b50afacfa411ed83cf0a54be41f961