**COMSATS University Islamabad,   
Abbottabad Campus**

**SOFTWARE DESIGN DESCRIPTION   
(SDD DOCUMENT)**

**for**

**SMS SPAM FILTERING**  
Version 1.0

***By***

**Waseem Ullah CIIT/FA20-BSE-129/ATD**

**Muhammad Adeel CIIT/ FA20-BSE-163/ATD**

**Sooraj Lal CIIT/ FA20-BSE-004/ATD**

***Supervisor*Supervisor Name: Muhammad Ali Faisal**

***Bachelor of Science in Computer Science (2020-2024)***

**Table of Contents**

**Revision History 3**

**1.** **Introduction 5**

**2.** **Design Methodology and software process model 5**

**3.** **System Overview 5**

3.1 Architectural Design 5

3.2 Process Flow/Representation 5

**4.** **Design Models [along with descriptions] 5**

**5.** **Data Design 6**

**6.** **Algorithm & Implementation 6**

**7.** **Software Requirements Traceability Matrix 6**

**8.** **Human Interface Design 7**

**9.** **Appendix I 7**

**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason for changes** | **Version** |
|  |  |  |  |
|  |  |  |  |

**Application Evaluation History**

|  |  |
| --- | --- |
| **Comments (by committee)**  **\*include the ones given at scope time both in doc and presentation** | **Action Taken** |
|  |  |
|  |  |

**Supervised by**

**<Supervisor’s Name>**

Signature\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Introduction**

Our project, SMS Spam Filtering, tackles the issue of annoying and potentially harmful SMS messages. We've completed 10% of the presentation in the Project Feasibility stage, covering modules like Authentication (Sign-Up, Log-In, Reset Password) and the Home Page (categorizing Spam/Block and Non-Spam Messages).

**Design methodology and software process model**

* **Design Methodology:** We're using a Procedural approach, focusing on step-by-step instructions. It aligns well with the React Native framework, emphasizing functional programming.
* **Software Process Model:** We've chosen an Agile methodology, allowing us to work in small, flexible steps, adapt to changes quickly, and collaborate effectively.

**System overview**

**Functionality:**

Our system improves user privacy and security by quickly sorting SMS messages into spam or non-spam based on content, sender reputation, and user choices.

**Key Functionalities:**

* Message Sorting
* Real-Time Detection
* User Customization
* Scalability
* Continuous Improvement
* Privacy Compliance

**Modules:**

1. Authentication (Sign-Up, Log-In, Reset Password)
2. Home Page (Spam/Block Messages, Non-Spam Messages)

**Design Approach:**

Our design mostly follows a Procedural approach, making it easy to follow step by step. We integrate some functional programming aspects in React Native for better efficiency.

**Scope:**

The project deals with challenges like changing spam content, language limitations (optimized for English), and resource constraints while keeping it simple and effective.

**Architectural design**

The architectural design of the SMS Spam Filtering system follows the Model-View-Controller (MVC) pattern, providing a modular structure that separates concerns and enhances maintainability. Here's a high-level overview of how the system's modules collaborate:

**1. Model:**

* **Responsibility:** Manages data and business logic.
* **Components:**
  + Spam Detection Engine
  + User Preferences
  + Message Categorization Logic
* **Relationships:**
  + Communicates with the Controller to receive user input and update the View.

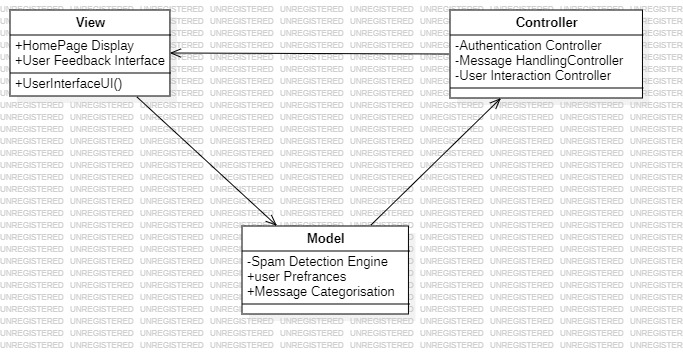
**2. View:**

* **Responsibility:** Presents information to the user and collects user input.
* **Components:**
  + User Interface (UI)
  + Home Page Display
  + User Feedback Interface
* **Relationships:**
  + Receives input from the user and sends it to the Controller for processing.
  + Listens for updates from the Model to refresh the UI.

**3. Controller:**

* **Responsibility:** Manages the flow of data between the Model and the View, processing user input and updating the Model accordingly.
* **Components:**
  + Authentication Controller
  + Message Handling Controller
  + User Interaction Controller
* **Relationships:**
  + Receives user input from the View and invokes appropriate actions in the Model.
  + Updates the View based on changes in the Model.

**Diagram:**



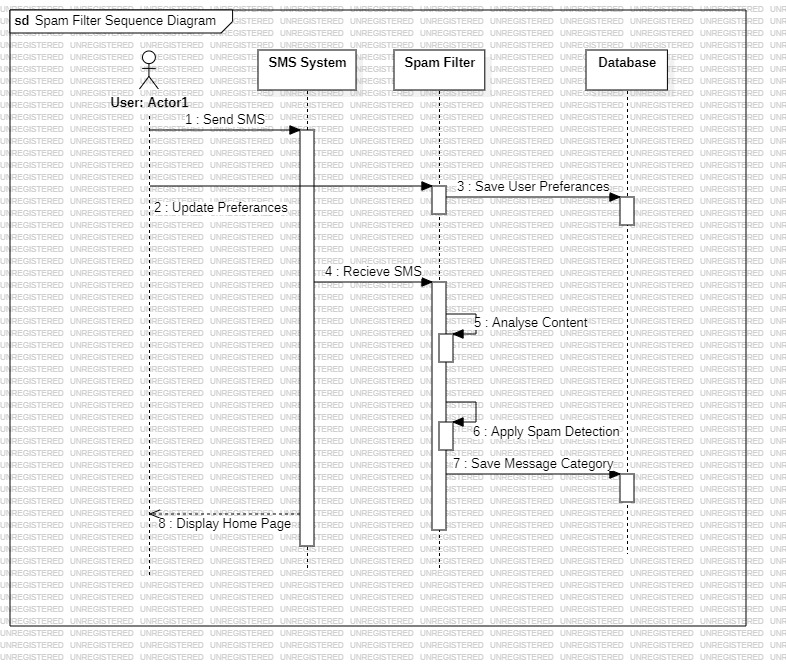
**Process flow/Representation**

A diagram of a flowchart

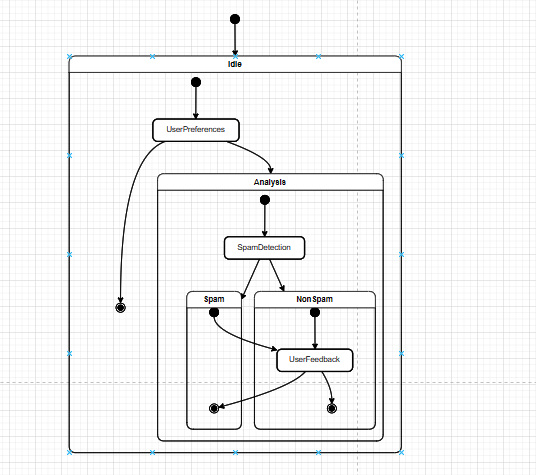
Description automatically generated

**Design models:**

**Sequence Diagram:**



**State Transition Diagram:**



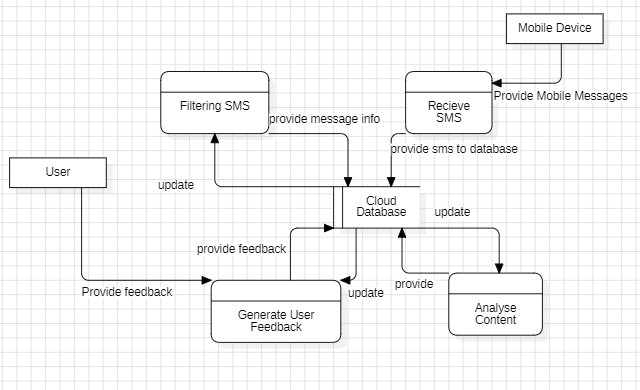
**Data Flow Diagram:**

**DFD LEVEL-0:**

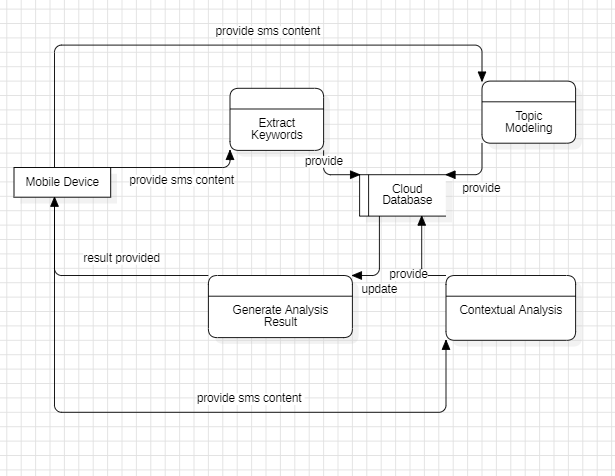
A diagram of a spam filtering

Description automatically generated

**DFD LEVEL-1:**



**DFD LEVEL -2:**



**Data design**

We’re using the Firebase Database management system by google. At the run time the data will be retrieved from the cloud database

**Algorithm & Implementation**

**Module 1: Authentication**

**Function 1: Sign-Up**

* Input: User details (username, password)
* Output: Success message or error
* Validate input
* Check username uniqueness
* Hash password
* Store details in the database
* Return success or error

**Function 2: Log-In**

* Input: User credentials (username, password)
* Output: Authentication token or error
* Retrieve user details
* Compare hashed password
* Generate token on success
* Return token or error

**Function 3: Reset Password**

* Input: User credentials (username, current password, new password)
* Output: Success message or error
* Authenticate user
* Update password if authenticated
* Return success or error

**Module 2: Home Page**

**Function 1: Spam/Block Messages**

* Input: SMS message
* Output: Categorized message (spam or non-spam)
* Analyze content using rules and ML
* Check sender reputation and preferences
* Categorize message
* Prevent spam from reaching the inbox
* Return categorized message

**Function 2: Non-Spam Messages**

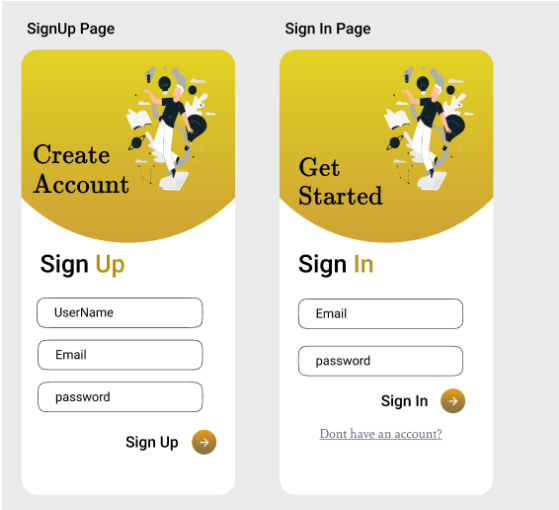
* Input: User preferences
* Output: List of non-spam messages
* Retrieve non-spam messages based on preferences
* Allow user review and management
* Report false positives/negatives
* Incorporate feedback for model improvement
* Return list of non-spam messages

**Software requirements traceability matrix**

|  |  |  |  |
| --- | --- | --- | --- |
| **Req. Number** | **Ref. Item** | **Design Component** | **Component Items** |
| FR01 | DFD | Level-0 | Overall system data flow |
| FR02 | DFD | Level-1 | Detail Process Data Flow |
| FR03 | DFD | Level-2 | Detailed data flow for Mock Interview Process |

**Human interface design**

**SignUp & SignIn Page:**



**Automatic Spam Filtering & Spam Protection:**

A screenshot of a phone

Description automatically generated