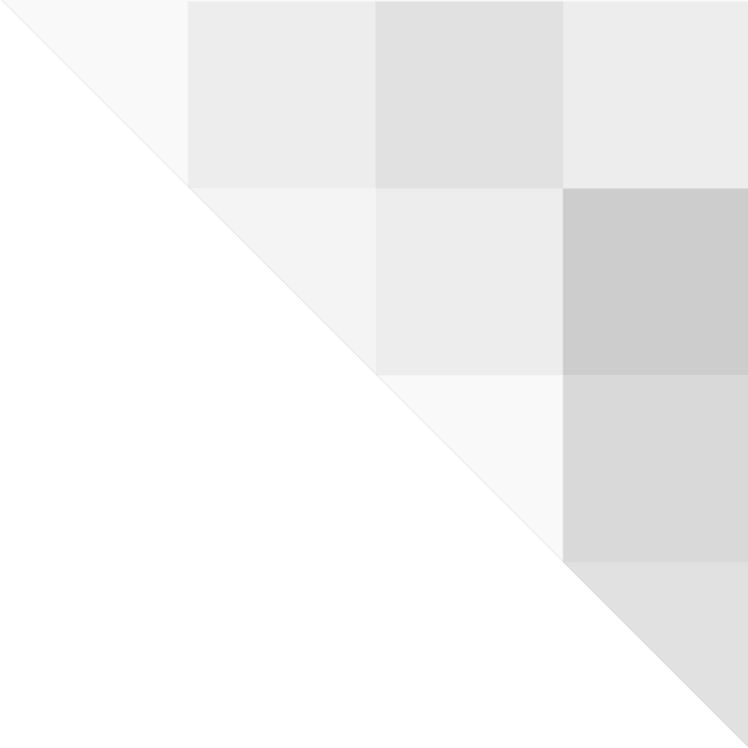
MedID

**Group 5**

FinalReport

Team Members

* Anmol Kumar, 20114042
* Arya Anand, 20114017
* Deepak Tailor, 20114029
* Dhruv Shandilya, 20114034
* Nikhil, 20114063
* Nishchay Narayan B, 20114065

**GitHub Repository :** <https://github.com/dhruv-shandilya/MedID.git>

# Project Proposal:

The app collects information about medical history of patient and ranks the medical reports on the basis of their seriousness.

It builds a connection between a doctor and patient by which all the medical reports can be accessed by the registered doctor or any medical institution.

For serious diseases like heart , diabetes, Blood Pressure related problems , lung diseases, liver diseases, kidney diseases, cancer, fatal viral and bacterial diseases the app reminds the patient for regular check ups within a specific time period

The app automatically highlights the allergies that a patient has for example allergy specific chemical compounds used in making medicines.

# Technology Used

* Android studio
* Java/Dart/Kotlin
* 8-16GB RAM
* Flutter
* AVD
* NVIDIA Card

# Key Features

We are making a “Medical Information App” which stores all the required information of patients through a particular ID like Aadhar Card. This app will be made publicly, any hospital can use it to store the patient’s information. We can say that this product is to be released to the open market. Just like how Aadhar card works, a medical ID number can be assigned to users which can be used by medical professionals to get timely access to medical information of the user in case of any mishap. Further building upon this idea, the user can save medical information of his/her illness as a file and this information can be accessed by a software which is exclusively present with hospitals. This will reduce the response time doctors need to read the medical reports and help determine what illness the patient has so that he/she can be given quick medical attention.

* **Ease of Access :** Easily accessible medical information at any time and place
* **Monitoring of patients:** Doctors can monitor patients efficiently
* **Updated Feedback and Review:** Survey of the users and resolving their queries and feedbacks

# Design Diagram

* **Flow Chart**

Diagram

Description automatically generated

* **Use Case Diagram**

Text

Description automatically generated

* **Class Diagram**

Diagram

Description automatically generated

# Description of Implementation

## 1. Development Process

We will be using Iterative Waterfall model because of the following reasons:

* Our software is neither too easy nor complex.
* It is well suited for well understood programs. Our goal for making this software is clear.
* As our project is small type software, we don’t need any kind of running prototype model.
* Our requirements are defined correctly at beginning of the project.

.

## 2. Outline Plan

* **Milestone 1 (14th Feb- 3rd March ) -** Learning to integrate different tools for our project
* **Milestone 2 (16th March – 22nd March) -** Feasibility Study Report
* **Milestone 3 (16th March – 22nd March) -** Requirements Analysis Report
* **Milestone 4 (22nd March - 6th April ) -** Software Design Phase
* **Milestone 5 (6th April- 14th April) -** Development/coding Phase First 7 days half of the team (front-end team) will work on the user interface of the website simultaneously the back-end team will work on creating the server. The other 7 days back-end team will work on connecting databases and APIs to the website and UI team will work on the rest of interface designing.
* **Milestone 6 (13th April – 16th April ) -** Deployment and Testing of the project.

**3. Coding and Implementation Phase**

* **Frontend**

The main application is programmed in android studio with flutter.

○ The frontend part is implemented in the dart language running in the flutter.

○ Image carousel is used as running images ○ Black theme is kept for the interactive interface.

○ Different colours and icons are used for every new interface for doctor and the patient.

* **Backend**

The backend consist of mainly 5 parts to be done:

**1. Authentication**:

Our website includes two types of authentication one with firebase\_auth which includes the usage of passport.js library for authenticating new and registered users and mongoose library for storing new users’ data and accessing registered users’ data. The other type of authentication is firebase\_auth which uses the Google+ API to access google account’s data of the users, this makes the user easy to signup/signin onto our website as they do not require to enter any details about themselves. Our backend directly takes the data from the Google response which is a json token which contains details like id, mobile number, etc.

* **Working of firecore:**

It requires a **`passport-local`** library. The user needs to give the details specified in our form, this data goes to the backend server with POST request which has the passport function to authenticate users’ data with our database(i.e sign up the user if not created till now or else log him in if already registered).

* **Working of GoogleStrategy:**

It requires a `passport-google-oauth20` library. The Google OAuth 2.0 authentication strategy authenticates users using a Google account and OAuth 2.0 tokens. The strategy requires a verify callback, which accepts these credentials and calls done providing a user, as well as options specifying a client ID, client secret, and callback URL.The Client Id and Client Secret needed to authenticate with Google can be set up from the Google Developers

Console(which is already set by our developers).Here we use the **env file** to protect the clientID and clientSecret. The below image shows the flowchart of Google OAuth20.

### 4. Database Management

We have used firebase database and these are the crucial things needed to be handled in database:

**Creating models**:

Creating models means creating different type of classes in database,for storing all the kind of data.We have basically created 8 models:

* + doctor
  + patient
  + licence id
  + medical reports
  + allergies
  + nurse

## 4. Integrating frontend, server, database and API

* Our frontend is written using dart in flutter with android studio**.**
* Databases used are firebase database.

So, we need to integrate all this. So, for that first all the requests made by frontend need to be responded to by our server.

## 5. Security Details

One of the major security threat is that the important medical details of the patient can get leaked. Medical history of a individual can be considered as private information that shouldn’t be accessible to others. So to avoid that we will make sure that nobody can access data of any particular user except the doctor and the concerned patient. We will have to apply strong encryption on all medical details. Thus we make sure there isn’t any security issue.

# Application of the project

# We are making a “Medical Information App” which stores all the required information of patients through a particular ID like Aadhar Card. This app will be made publicly, any hospital can use it to store the patient’s information. Just like how Aadhar card works, a medical ID number can be assigned to users which can be used by medical professionals to get timely access to medical information of the user in case of any mishap. Further building upon this idea, the user can save medical information of his/her illness as a file and this information can be accessed by a software which is exclusively present with hospitals. This will reduce the response time doctors need to read the medical reports and help determine what illness the patient has so that he/she can be given quick medical attention.

# Innovation in the Project

* Easily accessible medical info at any time and place
* Less response time in case of medical emergency
* Well interactive app that is user friendly
* Any allergies of user to a particular drug is alerted
* Doctors can monitor patients easily and at any time