# Implementation Document

## for

# Health Center Reimbursement

**Version 1.0**

**Prepared by**


**Group #: 6** **Group Name:** Quarantined Developers

| | | |
|---|---|---|
| Gaurav Kumar | 190337 | gauravkg@iitk.ac.in |
| Rahul Rathod | 190667 | rrathod@iitk.ac.in |
| Subhrojyoti Chatterjee | 190866 | sjyoti@iitk.ac.in |
| Shantanu Thakur | 190792 | sthakur@iitk.ac.in |
| Priyanka Jalan | 190649 | prianka@iitk.ac.in |
| Shorya Kumar | 190818 | shoryak@iitk.ac.in |
| Sanchit Sinha | 190753 | sanchits@iitk.ac.in |
| Atreya Goswami | 190201 | atreya@iitk.ac.in |
| Harsh Kumar | 190360 | harshku@iitk.ac.in |
| Nakula Neeraje | 190525 | nakula@iitk.ac.in |


**Course:** CS253

**Mentor TA:** *Nikhil Kumar Singh*

**Date:** 20/03/2022

# Revisions

| Version | Primary Author(s) | Description of Version | Date Completed |
|---|---|---|---|
| Initial Draft (v1) | Gaurav Kumar, Nakula Neeraje, | Built a web app that will allow users to apply for reimbursement claims online by simply filling an online form and providing required details. The | 00/00/00 |

| | | | |
|---|---|---|---|
| | Rahul Rathod, Shantanu Thakur, Shorya Kumar,Sanchit Sinha Subhrojyoti Chatterjee, Priyanka Jalan, Harsh Kumar, Atreya Goswami | system would also help users view already submitted forms and their status. The HC staff can scrutinize and verify the applications. They can approve and send the details to the Accounts department or reject the claims and update the status of the form.. | |

# 1 Implementation Details

*Provide the details of programming languages, frameworks, libraries, database systems, build systems, etc. that you have used for implementing your software.*
*Provide a brief justification of choosing any tool by stating its benefits over the alternatives.*

For the frontend part of the project, we have used HTML, CSS and Javascript. Advantage of using these languages is that they are supported by all browsers and they can be integrated with any backend framework. Also, they are lightweight and thus make the system faster.

For styling purposes, we used Bootstrap and Google Fonts. The advantage of using these frameworks in the frontend is that they help in creating responsive designs much faster than without them. They also provide cross-browser compatibility.

For the backend, we used Python with the Django framework. One of the biggest advantages of using the Django framework is that it is independent and complete in itself. It does not require any external solution. Django is everything from ORM to the web browser itself. Also codes in Django Framework are ‘Keep It Short and Simple’ and ‘Don’t repeat yourself’ compliant. Moreover, python also opens up the possibility to use machine learning or AI based components in the project.

As a database language we chose sqlite. Reasons for using this database language in place of others like MySQL and PostgreSQL are that sqlite is lightweight and thus provides better performance, does not require installation, is reliable and portable.

For version control and collaboration we used git and github respectively, github made communication between the team members very easy as all the contributors could review the pull requests locally and point out bugs and issues in the implementation and suggest changes that can be made, in a particular implementation. It made technical discussion very smooth and easy.

# 2 Codebase

*Provide the link to your github repository.*

***https://github.com/shoryak/hc_reimbursement ( link to the repository)***

*Mention briefly how to navigate the codebase.*

The directory structure of the codebase-

```
.
├── README.md
├── requirements.txt
└── src
    ├── db.sqlite3
    ├── dummyData
    │   ├── db.json
    │   ├── readme.md
    │   └── user_password.txt
    ├── manage.py
    ├── src
    │   ├── asgi.py
    │   ├── __init__.py
    │   ├── settings.py
    │   ├── urls.py
    │   ├── views.py
    │   └── wsgi.py
    └── user
        ├── admin.py
        ├── apps.py
        ├── __init__.py
        ├── migrations
        │   ├── 0001_initial.py
        │   ├── __init__.py
        ├── models.py
        ├── static
        │   ├── css
        │   │   ├── bootstrap.css
        │   │   ├── bootstrap.css.map
        │   │   ├── bootstrap.min.css
        │   │   ├── bootstrap.min.css.map
        │   │   ├── bootstrap-theme.css
        │   │   ├── bootstrap-theme.css.map
        │   │   ├── bootstrap-theme.min.css
        │   │   ├── bootstrap-theme.min.css.map
        │   │   └── index.css
        │   ├── fonts
        │   │   ├── glyphicons-halflings-regular.eot
```

```
│   │   ├── glyphicons-halflings-regular.svg
│   │   ├── glyphicons-halflings-regular.ttf
│   │   ├── glyphicons-halflings-regular.woff
│   │   └── glyphicons-halflings-regular.woff2
│   └── js
│       ├── bootstrap.js
│       ├── bootstrap.min.js
│       ├── jquery-3.5.1.min.js
│       └── npm.js
├── templates
│   ├── accounts_dashboard.html
│   ├── accounts_profile.html
│   ├── doctor_dashboard.html
│   ├── doctor_profile.html
│   ├── form.html
│   ├── hcadmin_dashboard.html
│   ├── hcadmin_profile.html
│   ├── index.html
│   ├── login.html
│   ├── patient_dashboard.html
│   ├── patient_profile.html
│   ├── signin.html
│   ├── signup_admin.html
│   └── signup.html
├── tests.py
├── urls.py
├── utils.py
└── views.py
```

./manage.py- Script that helps with management of the site i.e. making migrations and running the server

./dummyData/db.json- Stores dummy data that can be loaded for testing the application

./db.sqlite3- Database in sqlite3, storing data in tables based on different models defined in model.py

./user/settings.py- Contains the configuration of our website, i.e. database settings, logging configuration and location of static files.

./user/model.py- Stores all the models for our project i.e. an object which is stored in the database. The properties and methods for each object are defined in the class.

./user/admin.py- Manages the Django admin interface by registering the models defined.

./user/views.py- Contains the main logic of the application which requests information from the models created and pass it to the template

./user/urls.py- All the paths for the views and to access the application are configured.

./user/templates- Contain the frontend files to present the site in HTML.

./static- Static css and bootstrap files

The application can be tested by following the final procedure-

1) A virtual environment is create using the following command-
   ```
   python -m venv myvenv
   ```
   which needs to be activated using
   ```
   myvenv\Scripts\activate
   ```
2) The software requirements for the given application are installed using-
   ```
   pip install -r requirements.txt
   ```
   which installs dependencies like django
3) The database is created and the migrations are applied using-
   ```
   python manage.py migrate
   ```
4) To login as admin, a superuser account is created which has control overall control of the site by running-
   ```
   python manage.py createsuperuser
   ```
5) The web server is started by running-
   ```
   python manage.py runserver
   ```

# 3 Completeness

*Provide the details of the part of the SRS that have been completed in the implementation.*

*Provide the future development plan by listing down the features that will be added in the (may be hypothetical) future versions.*

## Implemented Features

- **Login (All Roles):** Created a login page which redirects the user to the profile dashboard. It also displays the corresponding errors if incorrect/nonexistent credentials are entered.
- **Patient Sign Up:** Login page comes with a signup button to create a user with a patient role, if they are not already enrolled. The signup page takes user details and user given password to create a user profile with a patient role.

- **User Profile Page (All Roles):** Each role has its own dashboard which has the profile button. This button will redirect the user to their corresponding profile page. Profile page contains different user info and also options to update certains features (details below).

- **Update Profile:** The rofile Page contains options to update the details selectively which can change over time.
    - **Patient Profile Update**: Patient can update/add details: Contact Info, Address, Bank Details (Bank Name, Bank IFSC Code, Bank Account No.), in their profile.
    - **Doctor Profile Update**: Doctor can update/add details: Contact Info, Address & Specialization, in their profile.
    - **HC Admin Profile Update**: HC Admin can update/add details: Contact Info, Address, in their profile.
    - **Accounts Profile Update**: Accounts users can update/add details: Contact Info, Address & Specialization, in their profile.

- **User Dashboard** (Role Wise Different Dashboard): After login, the user is redirected to their role-based dashboard page. The dashboard is the home menu after login. The dashboard page contains information about different reimbursement transactions (details below) as per accessible according to the roles. Roles can also View Form (details below) & View Transaction (details below) for each transaction. Also, allow users with any roles to Comment (details below) on a Transaction.

    - **Patient Dashboard:** Allow users with Patient role to view all the previous/current Reimbursement Transactions created by the user. Also contains options to Create a New Transaction (Fill Form) & Check Profile (and Update).

- **HC Admin Dashboard:** Allow users with HC Admin role to view their profile (and update) & shows all the Reimbursement Transactions Details claimed by patients.
  Five views:
  - Pending Forms: Contains New Claims' details created by patients not yet reviewed by HC Admin.
  - Waiting for Doctor Approval: Contains Claims' details passed on for Doctor Approval and yet waiting for their response.
  - Arrived after Doctor Approval: Contains Claims details which are verified by the mentioned Doctor and came back for HC Admin Approval to pass on to the Accounts next.
  - Approved Forms: Contains claims accepted by the HC Admin.
  - Rejected Forms: Contains claims rejected by the HC Admin.

- **Doctor Dashboard:** Allow users with Doctor role to view their profile (and update) & shows the corresponding Reimbursement Transactions Details assigned to this doctor user.
  Three views:
  - Pending Forms: Contains New Claims Details in which this medical advisor is chosen in the form, after being accepted by HC Admin Once.
  - Approved Forms: Contains claims accepted by this Doctor.
  - Rejected Forms: Contains claims rejected by this Doctor.

- **Accounts Dashboard:** Allow user with Accounts role to view their profile (and update) & shows all the Reimbursement Transactions Details assigned to this accounts user.
  Three views:
  - Pending Forms: Contains new claims details which are passed on by the HC after the doctor's approval.
  - Approved Forms: Contains claims accepted by this Accounts Officer.
  - Rejected Forms: Contains claims rejected by this Accounts Officer.

- **Admin Sign Up (For Admin Roles):** The HC Admin User is given a Admin Sign Up button to create users with Admin Roles i.e. Doctor, HC Admin & Accounts roles. This is done so that only the admin has the right to create admin role users.

- **Create a Reimbursement Request:** Allows the user to put forth a reimbursement request along with the details of the reimbursement so that the data can be converted to transactions which are then sent for verification and approval.
  - **Fillup Form Details:** The fillup form contains the details : Form ID, Patient name, Medical Advisor's name, consultation date, consultation fee, number of paid visits.

    - **Choose Medical Advisor:** Allows the user to mention the doctor who prescribed the medicines and tests.
    - **Choose/Add Medicines:** Allows the user to add the prescribed medicines by choosing the medicines from the medicine database. The database contains the fixed prices of the prescribed medicine.
    - **Choose/Add Lab Tests:** Allows the user to add the prescribed lab tests by choosing the lab test, and to enter the cost incurred for the lab test.
    - **Upload Documents:** Allows the user to upload the required documents such as doctor's prescription, medical bills; all together in a single merged PDF.
    - **Track Reimbursement Amount:** Allows the user to keep track of the amount that has been reimbursed and the amount that is yet to be reimbursed.

- **View Form:** Allows the user with any role to view the submitted reimbursement form of any reimbursement claim.

- **View Transaction:** The user with any role is allowed to view the reimbursement transaction details. This contains the details: Transaction ID, Form status, Reimbursement amount, Feedback, Date of transaction creation, updation and approval.

- **Feedback on a transaction:** Allows the user to provide feedback on the transaction which can be viewed in view transaction.

## Future Development Plans

- **Update UI:** Make changes in the UI to make the system more user friendly and fluid by providing intuitive animations and a uniform colour code to represent specific information without discrepancies.
- **Useful Links:** Provide a page with useful links related to HC. Provide a page with just info on Offline Reimbursement Plan.
- **Change Password Option:** Allows the user to change the password by providing the current password, and then asking for the new password and its retype. This changes the user password in the database.
- **Notification by Mail** (whenever Status Change or Feedback received): Whenever the reimbursement status changes, the patient user who filled the form is automatically notified through their mail.
- **Manage Notification:** Provides the option to toggle the notifications ON or OFF, or even selectively enable the notifications.
- **Provide a multiple feedback system:** A feedback system which allows users with any roles to discuss by multiple comments on a transaction.
- **Create a Draft Form**: Allows the user to temporarily save the form details without having to discard the already written data.
- **Categorized Document Upload:** Allows different categories for the documents, like prescription, medicine bill, lab test bill, etc.

- **Expand the user base to non-CC user IDs:** Allows the users with no CC user ID to have an alternate ID to register to the online reimbursement procedure.

# Appendix A - Group Log

<*Please include here all the minutes from your group meetings, your group activities, and any other relevant information that will assist in determining the effort put forth to implement your software*>

We all collaborated with each other in many meetings and discussed over various aspects of the projects over the period. The timeline for the project development can be viewed from the commit history of our Github Repository. We live shared and coded many parts together. We created PR requests, discussed them, tested the functionalities of the PR, suggested changes and also used them to update our Software and collectively developed our software. Our collaboration is visible in comments and reviews in PRs/issues created on GitHub.