
Implementation Document

for

Health++

Version 1.1

Prepared by

Group #: 5

Group Name: StarBugs

Name	Roll no.	E-Mail
Anuj	210166	anuj21@iitk.ac.in
Pulak Gautam	210791	pulakg21@iitk.ac.in
Dev Gupta	210318	devgupta21@iitk.ac.in
Abhimanyu Sethia	190023	sethia@iitk.ac.in
Deepanshu	210310	deepanshu21@iitk.ac.in
A. Atulya Sundaram	210001	aatulya21@iitk.ac.in
Kushagra Bhatnagar	210550	kushagrab21@iitk.ac.in
Pinkesh Mahawar	210719	pinkeshm21@iitk.ac.in
Pradeep Chahal	210733	pradeepc21@iitk.ac.in
Vaishnavi	211140	vaishnavi21@iitk.ac.in

Course: CS253

Mentor TA: Anuj Shrivastava (anujsh@cse.iitk.ac.in)

Date: 16 March 2023

CONTENTS	II
REVISIONS	II
1 IMPLEMENTATION DETAILS.....	1
2 CODEBASE	4
3 COMPLETENESS.....	5
APPENDIX A - GROUP LOG	7

Revisions

Version	Primary Author(s)	Description of Version	Date Completed
1.1	Anuj, A. Atulya Sundaram, Pulak Gautam	Describes the various frameworks used in our project, also gives an overview of our codebase and how to deploy it, along with the details of features that have/have not been implemented.	18/03/23

1 Implementation Details

1.1 Database Management System

SQLite is a popular open-source database management system that is widely used in many applications, including mobile apps, web browsers, and desktop software. It is a lightweight and fast database software that can run on various operating systems, including Windows, Linux, and macOS. Here are some of the advantages and benefits of using SQLite:

- Comes Integrated with Django: SQLite is the default database that is integrated in the Django framework and hence it allows integration of database with the implemented classes to be done very conveniently.
- Lightweight and Fast: Unlike traditional database management systems, SQLite does not require a separate server process, which makes it much lighter and faster. It can handle large volumes of data and perform complex queries with ease.
- Low Maintenance: SQLite requires minimal maintenance, as it does not have complex setup requirements or ongoing maintenance needs. Its simple design makes it easier to manage, update, and backup data.
- Cross-platform Compatibility: SQLite can run on various platforms, including Windows, Linux, and macOS. It is also compatible with many programming languages, including Python, Ruby, Java, and C++.
- ACID Compliant: SQLite is transactional, meaning that it can maintain data consistency and integrity in the event of system failures, power outages, or other unexpected events and it supports ACID (Atomicity, Consistency, Isolation, and Durability) transactions.
- Scalable: While SQLite is typically used for small to medium-sized applications, it can scale up to handle larger datasets and higher traffic loads. It supports concurrent access and can handle multiple users simultaneously.
- Security: SQLite offers robust security features, including encrypted database files, user authentication, and access controls. It also supports role-based access control (RBAC), which allows administrators to define roles and permissions for users.

Overall, SQLite offers many advantages and benefits over other database management systems like MySQL, MongoDB etc. Its simplicity, speed, and flexibility make it an excellent choice for our project.

1.2 Backend Framework

The Backend of our server is implemented using python, implementing the Django framework considering that it is easy to use and most of our group members had prior experience with python. Django is a popular open-source Python-based web framework that is widely used for building robust and scalable web applications. It provides a high-level development environment and

follows the model-view-controller (MVC) architectural pattern. Here are some of the advantages and benefits of using Django as a backend framework:

- Rapid Development: Django provides a high-level development environment that makes it easy to build complex web applications quickly. It comes with a built-in admin interface, authentication system, and other features that allow developers to focus on building their application's core functionality.
- Scalability: Django is highly scalable and can handle large traffic loads and high volumes of data with ease. It supports horizontal scaling, which means that multiple instances of the application can be run on multiple servers.
- Security: Django provides robust security features, including protection against common web vulnerabilities, such as SQL injection and cross-site scripting (XSS). It also provides a built-in user authentication system, role-based access control, and password management.
- Versatility: Django is a versatile framework that can be used to build a wide range of applications, including e-commerce websites, content management systems (CMS), social networks, and more.
- Compatibility: Django is compatible with many operating systems, including Windows, Linux, and macOS. It is also compatible with various databases, including SQLite, MySQL, and PostgreSQL.
- Modularity: Django is a modular framework, which means that developers can use only the components they need and customize them as required. This allows for greater flexibility and reduces development time.

Overall, Django provides many advantages over other popular frameworks like NodeJS and Express. Especially its ease of implementation makes it suitable for making projects in a short duration.

1.3 Frontend Framework

We have used Bootstrap library for designing the frontend of our project as it is one of the most popular open-source CSS Library because of the advantages that it provides to the developers. Bootstrap is a popular open-source frontend design framework that is widely used for building responsive and mobile-first websites and web applications. It provides a collection of pre-designed UI components, such as buttons, forms, tables, and navigation menus, that can be easily customized and integrated into a web project. Here are some of the advantages and benefits of using Bootstrap as a frontend design framework:

- Responsive Design: Bootstrap provides a responsive grid system that adjusts to different screen sizes, making it easy to build mobile-first websites that look great on any device.
- Cross-browser Compatibility: Bootstrap is compatible with all major web browsers, including Chrome, Firefox, Safari, and Internet Explorer. It also supports older versions of these browsers, ensuring that websites built with Bootstrap are accessible to all users.
- Customizable: While Bootstrap provides pre-designed UI components, it also allows developers to customize them to fit their specific design requirements. This makes it easy to create a unique look and feel for a web project while still leveraging the benefits of a pre-designed framework.
- Consistent Design: Bootstrap provides a consistent design language across all UI components, making it easy to maintain a unified look and feel throughout a web project.
- Time Saving: Bootstrap saves time and effort by providing a set of pre-designed UI components that can be easily integrated into a web project. This reduces the time and effort required to design and develop custom UI components from scratch.

Overall, Bootstrap provides many advantages and benefits over other frontend design frameworks, making it a popular choice for developers and businesses alike. Its responsive design, cross-browser compatibility, ease of use, customizability, consistent design, and time-saving features make it an excellent choice for building websites and web applications of any size or complexity.

2 Codebase

The complete integrated code for frontend as well as backend is available on GitHub at: <https://github.com/atulyasndrm/Starbugs.git>

Upon cloning the one will find the Health++ directory which contains all the source files. The directory contains install.sh and launch.sh which are the shell scripts which enable easy installation and launching of the website, by installing the dependencies and making the migrations required. It also contains HCapp, the directory containing all the source files of the software, implemented as a Django Project. In this directory, we find the manage.py file, a file required to work with the Django Project and its apps. The static folder contains the files required in the HTML to render the website's frontend. The media folder contains images of the prescription that will be uploaded when a student applies for a Leave. Since we are using SQLITE as our RDBMS, one will also find the db.sqlite3 file that contains our databases and tables. There is also the HCapp directory(again) which contains the configuration files for the Django project, where one can set all the URLs for the various apps and declare all the apps in the project.

The remaining folders contain the source files for the various django apps, used to enable all the functionalities of our Software. In each of these folders one finds the views.py file, containing the path redirections for the various HttpRequests and the rendering the required html pages containing the appropriate data , responding to various queries and validations. The urls.py contains the various urls for each application and contains the instructions for the function redirections. The templates in each app contains the html file to be rendered with the required Django templates. The models.py contains the classes and models of each of the users and the application. The rest of the files have not been used in our implementations.

3 Completeness

We were able to implement all the major functionalities completely mentioned in the SRS document. Brief description of the same has been mentioned below:

1. User Login (Sign Up/Sign In)
 - The application features a different user class for each proposed actor: Student, Doctor, Ambulance Driver, Receptionist and DUGC.
 - Any student can register and log in via the webpage directly, whereas the other users must approach the System Admin with a request for User Credentials allocation.
 - Users undergo process of authentication and designation of appropriate rights on each login.
 - Users differ in their authority to access the database and its functionalities.
2. Online Appointment Request
 - The application includes the implementation of an automated appointment system with an easy-to-use interface as per the SRS.
 - Patients can view and book available slots; Doctors can update their availability; Staff and Doctors can view the appointments schedule.
 - To book an appointment, user must fill the appointment form mentioning his/her symptoms and preferred type of doctor (if it a specialized treatment is sought after).
3. Appointment Allotment System
 - The system has an implemented appointment allotment system which redirects the patient's request to the sought-after doctor.
4. Ambulance Booking
 - Patients have the functionality to book an ambulance remotely.
 - Staff can also attend to ambulance requests via the application.
 - The requests are redirected in an orderly fashion to the ambulance drivers.
 - The software has been implemented in such a way that the requests do not clash for a particular ambulance driver.
5. Medical Leave Application
 - By attaching medical records with their application, students have the ability to create Leave Applications and monitor their status. The Leave Application can then be approved or declined by SUGC.
6. Blood Donation Facilities
 - Any user can raise a request for blood for a given blood group; notifications for the same will be broadcasted across all the users.
7. Campus Health Statistics
 - System admin can see macro statistics on the number of appointments sought, number of prescriptions given etc. and thereby gauge any spikes quickly.
 - The major numbers are also shown in the homepage to highlight ongoing activities and

There does persist flaws in our implementation giving room for improvement in our implementation in future versions. These have been mentioned as follows:

1. The application lacks a feature of viewing user credentials and updating the same.
2. It further lacks the functionality of updating password in case of password breach or if the user forgets his/her password.
3. The blood donation functionality does enable to open a blood donation request, but only the system admin has the ability to close irrelevant or old requests. This functionality can be extended to receptionists/staff.

Furthermore, in future versions we would also like to port our software base to mobile or light-weight desktop versions, enabling swift use of the software stack by the staff and ambulance staff.

Appendix A - Group Log

Date of meeting	Duration of meeting	Agenda of meeting
12 th February 2023	1hr	Discussed how the project will be implemented, decided that frontend and backend team (divided in the project plan) can work independently and then finally the frontend work will be integrated into the backend.
14 th February 2023	2hr	Discussed and explored the different frameworks that are popularly used for making such kind of project took some design decisions.
17 th February 2023	2hr 30mins	Initiated the work by making the project repository on GitHub, discussed the ways we can effectively collaborate. Also, figured out the resources from where we can get knowledge of the decided frameworks.
27 th February 2023	2hr	Met after the mid-sems to start working on the coding part. All of us had attained beginner knowledge about our frameworks.
1 st March 2023	1hr 30mins	Frontend team met for creating the webpages, divided in groups of 2 for each page and kept discussing our progress in order to maintain design consistency across different pages.
3 rd March 2023	2hr	The fundamental webpages in our project were ready, discussed the other pages that need to be created for certain features.
6 th March 2023	1hr	Most of the pages for the frontend were implemented, however we discussed areas where we can improve like animations, effects and backgrounds.
8 th March 2023	1hr	Backend team discussed how the classes are going to be implemented and learnt to use Django for maintaining database.
10 th March 2023	1hr 30mins	The whole team met to discuss how to proceed with the integration of frontend and backend.

Software Implementation Document for Group 5

13 th March 2023	2hr	Discussed some problems faced during the integration of frontend and backend and found ways to tackle those problems, this also required some changes in the frontend so that it can be easily integrated.
15 th March 2023	1hr	Brought the project to a completely working state, identified the few glitches and errors and started to fix them.