Software Engineering Lab
Implementation II - Build II

Virtual Clinic - An Integrated Care System

—

Mishal Shah - 16CO125

Samyak Jain - 16CO254

March 30, 2018

1. Basic Information

There was no change or any new addition to the functional requirements. Only the pending functional requirements, given in the below table were implemented in build-2. Also many non-functional requirements like performance requirements, safety,security, software quality attributes and data-backup were also implemented and taken care off.

2. Pending/ New Functional Requirements (Implemented)

FRID	NAME	Pending/new	Description
Req-3/3 Req-5/3	Delete Patient , Doctor, Admin, Lab Profile	Pending	Delete the patient profile if required by the admin
Req-4	Maintain Symptoms	Pending	This is a master list of Symptoms that can be selected by Patient during Consultation Request
Req-4/1	Add Symptom	Pending	Add a new Symptom
Req-4/3	Delete Symptom	Pending	View and Update the details of an existing Symptom i,e. description.Delete the Symptom Details - A Symptom can be deleted if and only if no Consultation request ever raised has used this symptom
Req-11	View/Generate Prescription/ Print Invoice	Pending	Patient shall have the facility to view the Prescription and print the invoice. Alternately, Admin can print and email the Prescription and Invoice
Req-9	Deliver Medicines	Pending	Chemist shall be able to view the open prescriptions assigned to them (based on Pincode) and have the facility to update the status. Medicines shall be delivered

|			offline to Patient by Chemist
Req-10	Perform Lab Tests	Pending	Lab In-charge shall be able to view open lab-requests and have the facility to update the status- Actual Sample Collection and lab tests shall be offline. Labs shall also submit reports to Patient offline
Req-12	Housekeeping Patient Profiles and Consultation Requests	Pending	Quarterly all Patient Profiles which have not raised any Consultation Request for last 3 years will be archived. Also, every quarter, Consultation Requests that are more than 3 years old shall be archived. This will improve the performance
Req-13	Daily Back-up	Pending	Every night, full back-up of the system (both Database and Code) shall be taken and sent to remote site for safekeeping. The back-ups shall be rotated every 30 days.

3. Functional Requirements (Not Implemented)

FRID	NAME	Description	Why not implemented in this build?
Req-7	Predict Disease	This is a batch program that will run periodically on new Consultation Requests to predict diseases based on machine learning Algorithm and prioritize the Requests (Critical/ Medium/ Low)	To keep the batch program running, we need a system having atleast 16 GB RAM, and a seperate VM through which we can directly provide read and write facility for the data, hosting the batch program on the Django server reduces it efficiency and also needed some configurations settings. We weren't able to figure out that also, our batch program wasn't giving the desired results so it wasn't possible for us to integrate
Req-16	Reports	The System will generate following reports: a) Doctor wise Appointment file for next 5 days b) Speciality wise - Number of Consultations by Doctor per month/ quarter/ year (run on last day of the month)	A part of it is implemented by us, by logging all the activity by each user, which can be seen in the profile of the admin. Though we haven't added the features of payment portal hence the revenue generated wasn't possible to be calculated. Also we aren't logging anything into the database, hence we

		c) Speciality wise - Revenue Earned per month/quarter/year (run on last day of the month) d) Patient History File - For a patient, history of consultations done in last 3 years.	couldn't query anything onto our database to get the desired results. The problem of using third party pdf generating service was also being faced here.

4. Screenshots of Functional Requirements Implemented in this Build

Req 3/3, Req 5/3, Req 12:
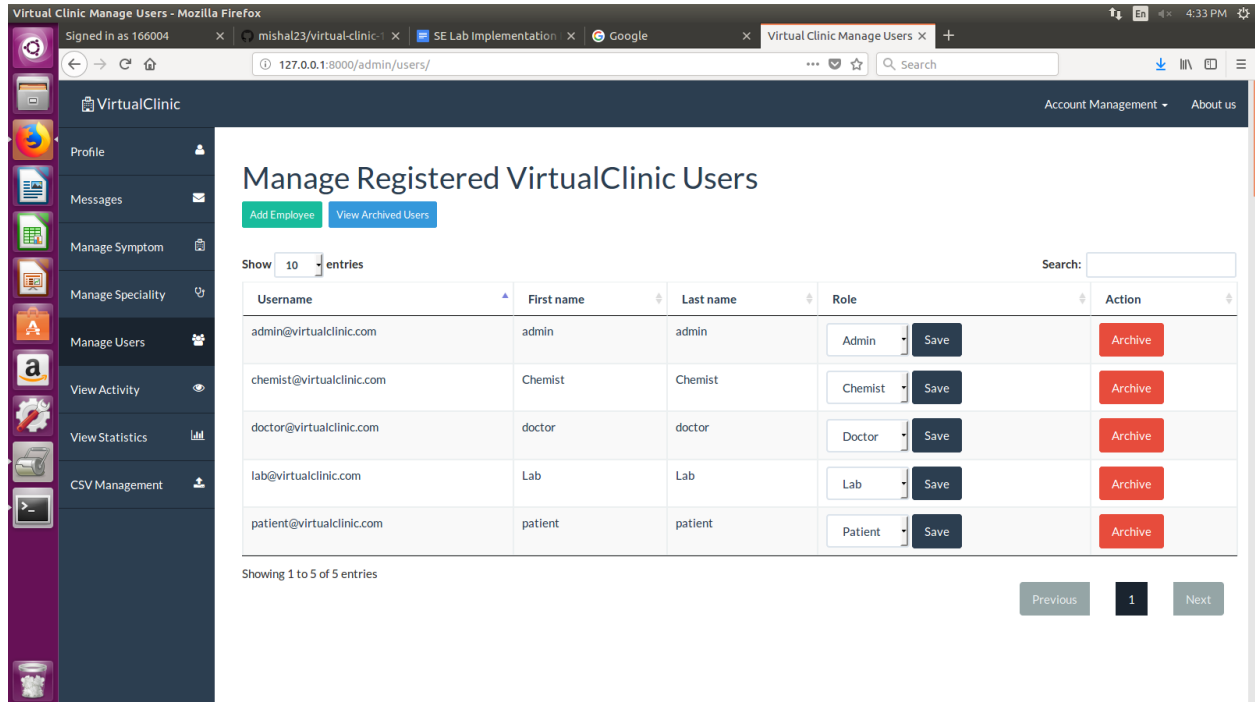
Delete(Archive) doctor/lab/chemist/patient/admin

Before deletion (archiving)

After deletion(archiving)

Req 4: Maintain Symptoms

Dashboard of maintain symptom as seen by admin

Req 4/1 : Add symptom

Before the symptom is added (Add symptom page)

After the symptom is added

Req 4/3:delete symptom

Before deleting symptom
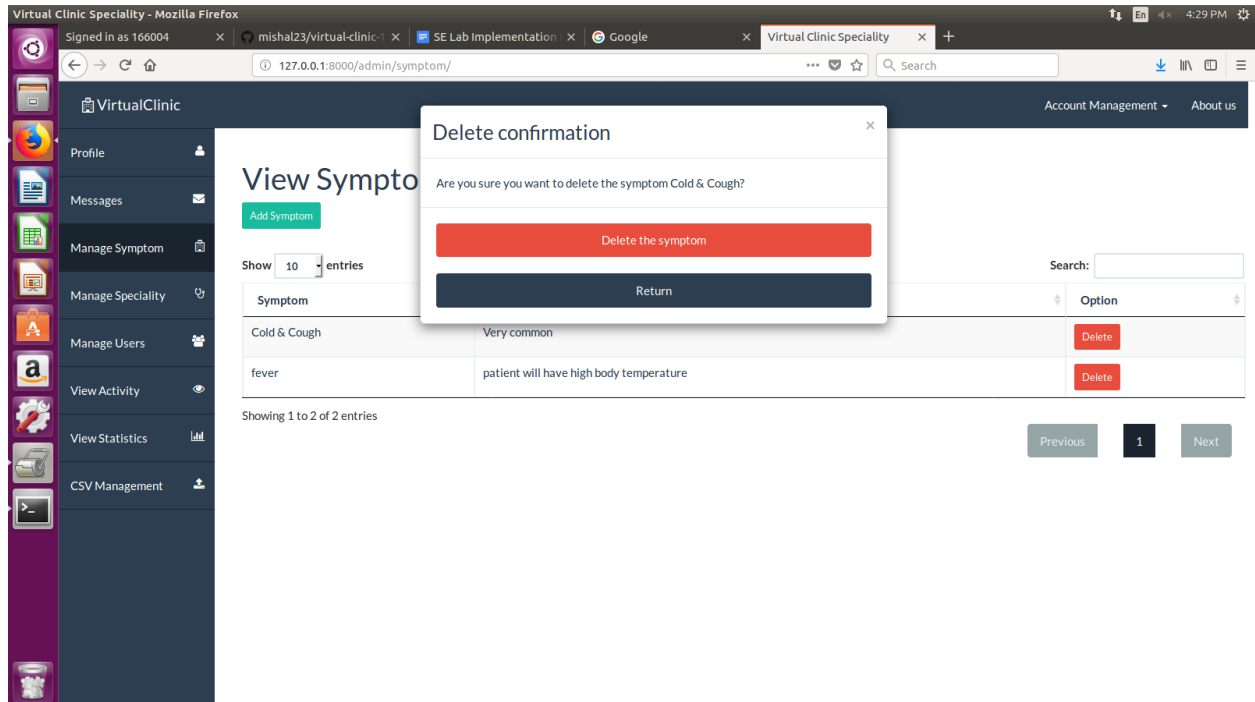
After deleting the symptom

Req 11: Generate invoice

Before generating the invoice (prescription page)

After generating the invoice (invoice generated as a pdf document)
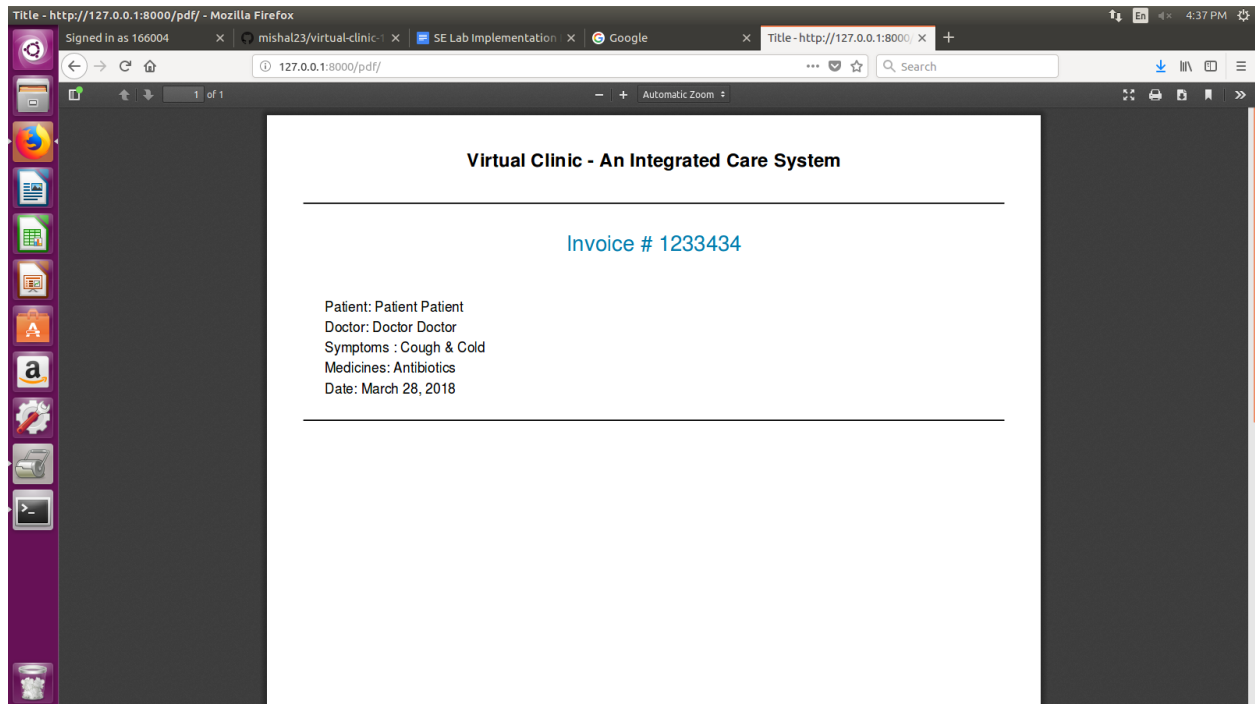
Req 9: Deliver Medicines

Chemist dashboard to view prescriptions

Chemist updates its delivery status

Req 10: Perform Lab Tests

Lab dashboard to display and update medical test

Upload Test results after performing the necessary test

Display test results

Req 12: Housekeeping patient profile and consultation Request

Before archiving the details (consultation request)

After archiving the details(consultation request)

NOTE: Housekeeping of patient profile was already implemented and shown in the first screenshot

5. Non-Functional Requirements Considered

NFRID	Name	Description
5.1	Performance Requirements	It includes variety of performance related requirements like robustness, response time, memory usage, load time, speed of the software etc.
5.2	Safety Requirements	The system shall be designed to comply with health care regulations as defined by Government of India.
5.3	Security Requirements	All communications between the browser and the website needs to be encrypted. Every stakeholder can login after giving the desired password and authentication.
5.4	Software quality attributes	The Key Software Quality Attributes are Availability, Reliability and Usability. As the system is expected to be 24/7 - High availability is very important. Also, since the major transaction i.e. Patient Consultation Requests has classified data/ information the system needs to be highly reliable.
5.5	Software back-ups and redundancy	All the day needs to be periodically backed up to ensure that the software runs efficiently even if there is a failure in the main system.

6. ScreenShots of Non-Functional Requirements

5.1 Performance Requirements

When deployed on server only, 2 requests of 22 fail in a minute

When deployed on server, approximate response time is 2 milliseconds to respond to the incoming request
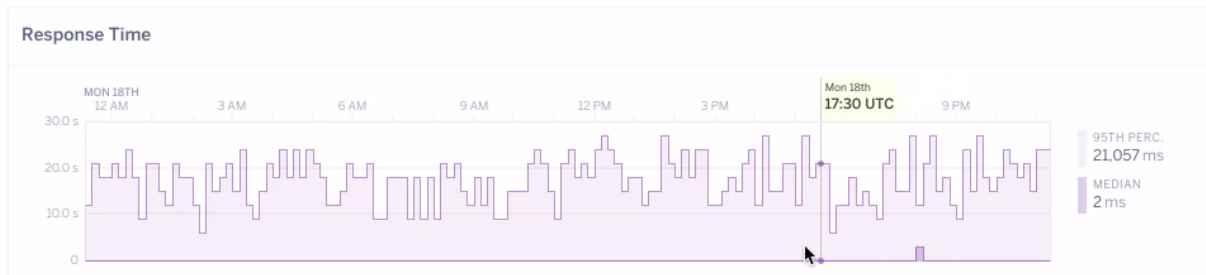
Memory Usage of the application:

Average Total : 67MB

Maximum Total : 67MB

Maximum RSS : 61MB

Maximum SWAP memory : 5MB
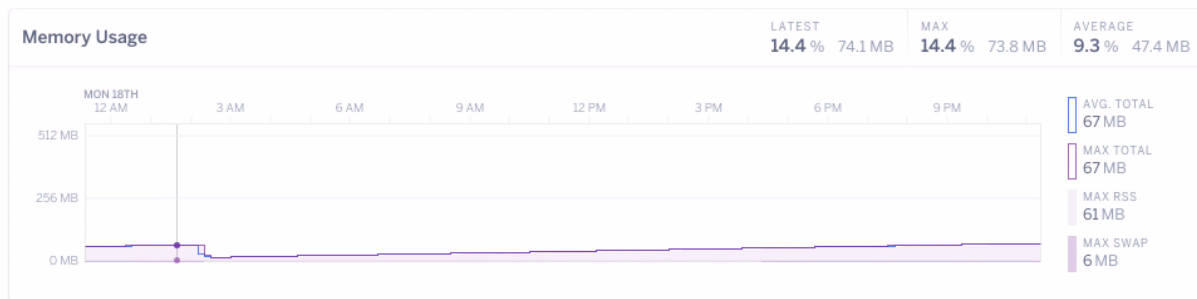
Upon login, the Data Object Model loading takes only 467ms, and the overall load time is 836ms.

5.2 Safety Requirements

Safety Requirements for each user are very properly taken care while implementing the software.

5.3 Security Requirements

The server is deployed over https which is the secure version of HTTP, the protocol over which data is sent between your browser and the website that you are connected to. The 'S' at the end of HTTPS stands for 'Secure'. It means all communications between your browser and the website are encrypted.
HTTPS is often used to protect highly confidential online transactions like online banking and online shopping order forms. Hence the security requirements are very well taken care off.

Moreover every stakeholder can access his/her account only by logging after the authentication.

5.4 Software Quality Attributes

As it can be seen, our software when deployed on server is available throughout the day
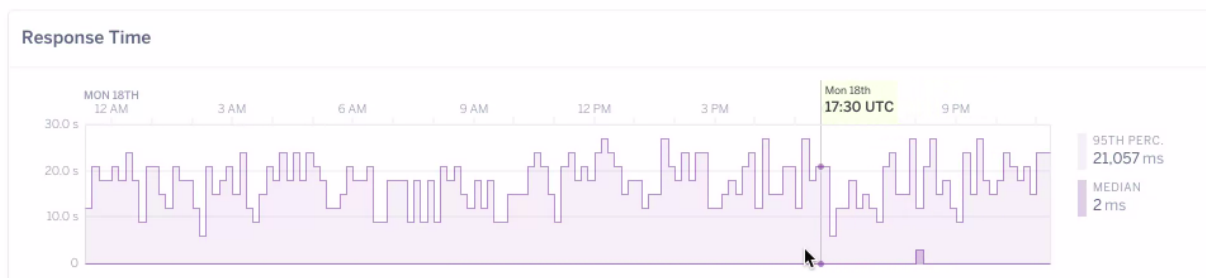
5.5 Software Backups & Redundancy:

As the data of each user is important, data backup is done.

Data Backup: All the data is periodically backed up in JSON format and stored in a slave database

Open ▾   Save

[{"model": "contenttypes.contenttype", "pk": 1, "fields": {"app_label": "admin", "model": "logentry"}}, {"model": "contenttypes.contenttype", "pk": 2, "fields": {"app_label": "auth", "model": "permission"}}, {"model": "contenttypes.contenttype", "pk": 3, "fields": {"app_label": "auth", "model": "user"}}, {"model": "contenttypes.contenttype", "pk": 4, "fields": {"app_label": "auth", "model": "group"}}, {"model": "contenttypes.contenttype", "pk": 5, "fields": {"app_label": "contenttypes", "model": "contenttype"}}, {"model": "contenttypes.contenttype", "pk": 6, "fields": {"app_label": "sessions", "model": "session"}}, {"model": "contenttypes.contenttype", "pk": 7, "fields": {"app_label": "server", "model": "location"}}, {"model": "contenttypes.contenttype", "pk": 8, "fields": {"app_label": "server", "model": "hospital"}}, {"model": "contenttypes.contenttype", "pk": 9, "fields": {"app_label": "server", "model": "profile"}}, {"model": "contenttypes.contenttype", "pk": 10, "fields": {"app_label": "server", "model": "account"}}, {"model": "contenttypes.contenttype", "pk": 11, "fields": {"app_label": "server", "model": "action"}}, {"model": "contenttypes.contenttype", "pk": 12, "fields": {"app_label": "server", "model": "appointment"}}, {"model": "contenttypes.contenttype", "pk": 13, "fields": {"app_label": "server", "model": "prescription"}}, {"model": "contenttypes.contenttype", "pk": 14, "fields": {"app_label": "server", "model": "medicalinfo"}}, {"model": "contenttypes.contenttype", "pk": 15, "fields": {"app_label": "server", "model": "medicaltest"}}, {"model": "contenttypes.contenttype", "pk": 16, "fields": {"app_label": "server", "model": "statistics"}}, {"model": "contenttypes.contenttype", "pk": 17, "fields": {"app_label": "server", "model": "message"}}, {"model": "contenttypes.contenttype", "pk": 18, "fields": {"app_label": "server", "model": "notification"}}, {"model": "contenttypes.contenttype", "pk": 19, "fields": {"app_label": "server", "model": "speciality"}}, {"model": "sessions.session", "pk": "0w26o3anxngt4ujzi00q0fjg9rh3r6nr", "fields": {"session_data": "MTJkOGY1MGFjZmViZDljYWYwZDM3MTEwNjQ5Mzk1MzllOGM2MzU1Nzp7Il9hdXRoX3VzZXJfaGFzaCI6IjEyNzA0Y2YyMmVjNDhlYzZiMjdhMmY0ODRjMmU0NTA3MDU1OWUyNzEiLCJfYXV...", "expire_date": "2018-03-25T15:55:30.878Z"}}, {"model": "sessions.session", "pk": "b89fm305otw338uexpw8y0etd7xfi6ri", "fields": {"session_data": "MzQ3OGYwNThlNzNlZmM4MmU3ZDhhOTRlOGZlNmU0YzQzMTZlYjIwNzp7fQ==", "expire_date": "2018-03-25T10:30:46.162Z"}}, {"model": "sessions.session", "pk": "bdtnz74pb11hmir25wi0z4v6z52w2ze9", "fields": {"session_data": "YmJlMDk1NTQ4ZjZlZWRjZDhhZjI2YjRiMDA0NjdhZGE1MjdmZTVmNTp7Il9hdXRoX3VzZXJfaGFzaCI6IjM0MDFkYTVjMTRkOTk2YjM5Nzk4NzI1ZjMxNjY1NTY0MzI0NmFjYjgiLCJfYXV...", "expire_date": "2018-03-25T17:24:46.305Z"}}, {"model": "sessions.session", "pk": "jabzdcl47814l085a6crkz80brh2uixo", "fields": {"session_data": "MzQ3OGYwNThlNzNlZmM4MmU3ZDhhOTRlOGZlNmU0YzQzMTZlYjIwNzp7fQ==", "expire_date": "2018-03-29T16:07:01.975Z"}}, {"model": "sessions.session", "pk": "sl0ahbzd48v0a2azvkgimtz6bxp00wcm", "fields": {"session_data": "ZTdiNTg3MTYwODI5YmMwN2UxNzJkMTQxYWIxNjM0NTcxYTI2ZDM3Yzp7Il9hdXRoX3VzZXJfaWQiOiIxIiwiYWxlcnRfc3VjY2VzcyI6IlNldHHVwIGhhcyBhbHJlYWR5IGJlZW4gY29tcGx...", "expire_date": "2018-03-18T10:12:27.025Z"}}, {"model": "sessions.session", "pk": "uz3042aveg2kiiog7b11mtwh1i2uw7ab", "fields": {"session_data": "MzQ3OGYwNThlNzNlZmM4MmU3ZDhhOTRlOGZlNmU0YzQzMTZlYjIwNzp7fQ==", "expire_date": "2018-03-20T11:06:48.274Z"}}, {"model": "sessions.session", "pk": "ytelw27ouu8bjcsqwupdrip3thv4nhfv", "fields": {"session_data": "NGU5Mjg3YzE2YmYzMThhODdhZDBkMWI4Njc4MDRmODA3ZWIzYTdmNjp7Il9hdXRoX3VzZXJfYmFja2VuZCI6ImRqYW5nby5jb250cmliLmF1dGguYmFja2VuZHMuTW9kZWxCYWNrZW5kIiw...", "expire_date": "2018-03-19T08:24:10.180Z"}}, {"model": "server.speciality", "pk": 1, "fields": {"name": "ENT", "description": "Ear Nose Throat"}}, {"model": "server.location", "pk": 1, "fields": {"city": "Unicorn", "zip": "12432", "state": "NY", "country": "India", "address": "1 Hope Street"}}, {"model": "server.location", "pk": 2, "fields": {"city": "Unicorn", "zip": "34232", "state": "NY", "country": "India", "address": "1 Happy Road"}}, {"model": "server.hospital", "pk": 1, "fields": {"name": "Memorial Strong", "phone": "453423421", "location": 1}}, {"model": "server.hospital", "pk": 2, "fields": {"name": "Highland", "phone": "4352534", "location": 2}}, {"model": "server.profile", "pk": 1, "fields": {"firstname": "admin", "lastname": "admin", "sex": "", "birthday": "1000-01-01", "phone": "", "allergies": "", "created": "2018-03-04T10:11:08.201Z", "prefHospital": null, "primaryCareDoctor": null, "speciality": ""}}, {"model": "server.profile", "pk": 2, "fields": {"firstname": "patient", "lastname": "patient", "sex": "", "birthday": "1000-01-01", "phone": "", "allergies": "", "created": "2018-03-04T20:05:41.350Z", "prefHospital": null, "primaryCareDoctor": null, "speciality": ""}}, {"model": "server.profile", "pk": 3, "fields": {"firstname": "doctor", "lastname": "doctor", "sex": "", "birthday": "1000-01-01", "phone": "", "allergies": "", "created": "2018-03-05T06:39:19.002Z", "prefHospital": null, "primaryCareDoctor": null, "speciality": ""}}, {"model": "server.profile", "pk": 4, "fields":

JSON ▾   Tab Width: 8 ▾   Ln 1, Col 1   ▾   INS

7. Summary of Test Plan

- All high priority functional requirements stated in SRS will be tested and the ones which are of low priority will not be tested in the testing phase of the project.
- Most of the functional requirements will be unit tested by unittest module.
- We will integrate Travis CI to our github repository so that all the tests are run and passed before anything is merged On it the production branch.
- Types of testing to be performed.
  - Functional Testing - BlackBox Testing
  - System Testing - Blackbox Testing
  - Performance Testing - Blackbox Testing
  - Usability Testing - Blackbox Testing
  - Acceptance Testing - Blackbox Testing
  - Beta Testing - Blackbox Testing
  - Unit Testing - Whitebox Testing
- Outcomes Expected
  - Functional Testing : We as developers of the software will be comparing our SRS and confirm that it is implemented and works correctly.
  - System Testing : For system testing, we will be installing our software on different Operating Systems and ensure that software works properly.
  - Performance Testing : We will be relying on a PaaS for speed and effictiveness of the system.
  - Usability Testing : This will be performed by the client to evaluate how user-friendly the GUI is? How easily can the client learn? After learning how to use, how proficiently can the client perform? How pleasing is it to use its design?
  - Acceptance Testing : This will be performed by the customer to ensure that the delivered product meets the requirements and works as the customer expects.
  - Beta Testing : This will be done by end users, a team outside development. This will cover some of the unexpected errors.
  - Unit Testing : This will be performed mainly by unittest module in python to ensure that all the functions are correctly implemented and working.

8. Summary

This report is divided into 7 sections each of them describing a different implementation aspect of the system. Section-1 gives the introduction about the functional requirements implemented. Section-2 depicts a table that shows the list of various functional requirement implemented in this build and states whether they were pending or new. Section-3 gives the list of functional requirement that could not be implemented in this build. Section-4 illustrates the functional requirements implemented in this build with the help of screenshots. Section-5 gives a list of Non-functional requirements considered while developing the system and is followed by the screenshots of the same. Last section gives the test plan i.e the next phase in the software development Life-cycle model.