Software Engineering Project

Project Group AnonCare

Software Quality Assurance Plan

version 1.0, 18 May 2016

AnonCare

Project Team:	Ma. Erikka M. Baguio Remarc E. Balisi Mary Grace P. Cabolbol Josiah Eleazar T. Regencia
Project Manager:	Ma. Erikka M. Baguio
Senior Manager:	Josiah Eleazar T. Regencia
Advisor:	Prof. Orven E. Llantos
Customer:	Mindanao State University - Iligan Institute of Technology Clinic

Mindanao State University - Iligan Institute of Technology

Abstract

In the AnonCare project, a system provides communication for all the employees in the university clinic to pass data from one employee to another. The software includes three basic users: doctor, nurse, and system administrator. All the users have their own profile. Nurse conducts examination to the patient. Nurse sends an appointment request to the doctor and sends the examination to the doctor. The doctor receives the examination and the generated initial diagnosis in order to make the final diagnosis. The system administrator will be in-charged of maintenance of the disease data together with symptoms and diagnostic questions.

Table of Contents

Document Status Sheet

Document Title	Software Quality Assurance Plan
Document Identification	Documents/Project/AnonCare/SQAP/0.1.0
Authors	Ma. Erikka M. Baguio Remarc E. Balisi Mary Grace P. Cabolbol Josiah Eleazar T. Regencia
Version	1.0
Document Status	draft

Version	**Date**	**Author**	**Summary**
0.0	2/5/16	Ma. Erikka M. Baguio Remarc E. Balisi Mary Grace P. Cabolbol Josiah Eleazar T. Regencia	Document Update
0.1	2/10/16	Ma. Erikka M. Baguio Remarc E. Balisi Mary Grace P. Cabolbol Josiah Eleazar T. Regencia	Update Chapter 1.
0.2	2/20/16	Ma. Erikka M. Baguio Remarc E. Balisi Mary Grace P. Cabolbol Josiah Eleazar T. Regencia	Update Chapter 1. Added Chapter 2.
0.3	2/23/16	Ma. Erikka M. Baguio Remarc E. Balisi Mary Grace P. Cabolbol Josiah Eleazar T. Regencia	Added Chapter 3. Added Chapter 4.

0.4	3/31/16	Josiah Eleazar T. Regencia	Updated Abstract Updated Chapter 1
0.5	3/31/16	Mary Grace P. Cabolbol	Updated Chapter 5
0.6	3/31/16	Ma. Erikka M. Baguio	Updated Chapter 6
0.7	4/2/16	Josiah Eleazar T. Regencia	Updated Chapter 7
0.8	4/2/16	Ma. Erikka M. Baguio	Updated Chapter 8
0.9	4/2/16	Remarc E. Balisi	Updated Chapter 9
0.10	4/4/16	Remarc E. Balisi	Updated Chapter 10
0.11	4/5/16	Josiah Eleazar T. Regencia	Updated Chapter 11
0.12	4/6/16	Mary Grace P. Cabolbol	Updated Chapter 12
0.13	4/6/16	Remarc E. Balisi	Updated Chapter 13
0.14	4/6/16	Ma. Erikka M. Baguio	Updated Chapter 14
0.15	5/13/16	Mary Grace P. Cabolbol	Updated All Chapters
0.16	5/15/16	Ma. Erikka M. Baguio	Updated Font sizes and indentation

Chapter 1

Introduction

1.1 Purpose

The purpose of this plan is to define the AnonCare Software Quality Assurance (SQA) organization, SQA tasks and responsibilities; provide the references and guidelines to perform SQA activities; and provide the standards, practices and conventions used in carrying the SQA activities; and provide the tools, techniques and methodologies to support SQA activities, and SQA reporting.

1.2 Scope

In the AnonCare project, a system has to be designed like a 3-door apartment where each basic user's access is only limited inside the forewalls of each apartment door. Thus, both nurse and system administrator are not authorized to view the medical records of a particular patient. At the same time, both doctor and nurse are also not authorized to add users to the system. Once examination is finished, the nurse gives the patient the option to choose which doctor the patient wants to be attended by, given both doctors of the clinic are present at the moment. The nurse then sends an appointment request together with the examination to the doctor. The requested doctor can reject the request and refer to the other doctor, given, the requested doctor thinks he/she is not fit for the situation of the patient.

1.3 List of Definitions

This section contains the definitions of all used terms, acronyms, and abbreviations in this document.

BDD	Behaviour Driven Development
Doctor	This is the person who will return the final diagnosis of the patient. This person has the ability to edit the examination if the doctor interviews the patients again.
Assessment	A process where the nurse get the name, vital signs, the complaint, etc.,
PM	Project Manager
SM	Senior Manager

MSU-IIT	Mindanao State University - Iligan Institute of Technology
Nurse	He/She conducts the diagnostic examination, sends requests for appointments and also sends examination to the doctor.
SQA	Software Quality Assurance
SQAP	Software Quality Assurance Plan
System Administrator	He has the authority to add patients, nurse, doctor, and diseases data.
SQA reporting	Describe the major quality assurance activities accomplished.
SR	Software Requirements
AD	Architectural Design
DD	Detailed Design
QAM	Quality Assurance Manager
QAE	Quality Assurance Engineer
Version Control Guidelines	1. Draft document version number. a. The first draft of a document will be version 0.1. b. Subsequent drafts will have an increase of “0.1” in the version number, e.g., 0.2, 0.3,... 2. Final document version number a. The first final version of a document will be Version 1.0. b. Subsequent final documents will have an increase of “1.0” in the version number (1.0,2.0, etc.). 3. Final documents undergoing revisions a. final Documents undergoing revisions will be Version X.1 for the first version of the revisions. While the document is under review, subsequent draft versions will increase by “0.1”, e.g., 1.1, 1.2, 1.3, etc. When the revised document is deemed final, the version will increase by “0.1” over the version being revised, e.g., the draft 1.3 will become a final 2.0

1.4 List of References

Reference materials used to develop the AnonCare SQAP includes:

[SQAP]	Software Quality Assurance Plan, SPINGRID team, Version 0.1.3, June 2006
[JESA]	Just Enough Software Architectures http://gotocon.com/dl/goto-cph-2012/slides/software-architecture/George-Fairbanks.pdf, G. Fairbanks, May 2012
[JUT]	JavaScript Unit Testing, Hazem Saleh, January 2013
[TP]	*Testing Python*: APPLYING UNIT TESTING, TDD, BDD, AND ACCEPTANCE TESTING, David Sale, 2104
[LPDP]	*Learning Python Design Patterns*: A practical and fast-paced guide exploring Python design patterns, Gennadiy Zlobin, November 2013
[PPTE]	*Python Penetration Testing Essentials:* Employ the power of Python to get the best out of pentesting, Mohit Raj, January 2015
[PWPTC]	*Python Web Penetration Testing Cookbook:* Over 60 indispensable Python recipes to ensure you always have the right code on hand for web application testing, Cameron Buchanan, June 2015

Chapter 2

Management

This section describes each major element of the organization that influences the quality of the software.

2.1 Organization

The project manager leads the anoncare team and assisted by the senior manager. The project manager should make sure they control risk and minimise uncertainty and he/she will be the one who is responsible for making decisions and every decision she will make must directly benefit the project. In any case, if complications arise, the project manager should be informed immediately as soon as possible. If the project manager is absent, the senior manager should be the one informed. It is then the job of the project manager or the senior manager to report to the doctor of the university clinic the complications faced along the way. Worst case scenario, the project manager must collaborate with the doctor of the university for necessary changes or remedies needed for problems in the system.

2.2 Tasks

This section contains the tasks to be done by the SQA team in the software life cycle covered by [SQAP].

- ➢ Evaluate software tools.
- ➢ Evaluate facilities
- ➢ Evaluate project planning.
- ➢ Evaluate system requirements analysis process
- ➢ Evaluate system design process.
- ➢ Evaluate software implementation and unit testing process.
- ➢ Evaluate end-item delivery process.
- ➢ Evaluate corrective action process.
- ➢ Verify project reviews.

The main task of the SQA team is to check whether the procedures are followed properly and that standards are handled correctly as defined in the [SQAP]. Additionally the SQA team inspects whether all group members fulfill their tasks according to the parts of the [SQAP] applying to their specific tasks.

2.3 Responsibilities

It the sole responsibility of every member of the Quality Assurance team and Development team to maintain the Quality of the project by following agreed conventions to the last dot. All Quality Assurance Engineers are encouraged to suggest styles and conventions but it is up to the Quality Assurance Manager to decide which to use. The roles and responsibilities are as follows:

<u>Quality Assurance Manager(QAM)</u>

Decides the conventions, styles of the Development project and the Quality Assurance team to use. The QAM also hands out tasks to each of the Quality Assurance Engineers(QAE) for them to implement. The QAM sees throughout the whole top-level view of the Quality Assurance Project.

<u>Quality Assurance Engineer(QAE)</u>

Sees throughout the details of their tasks handed to them by the QAM. The QAE follows the road plan given to them by the QAM. Should a QAE find a better solution than the proposed solution, the QAE should first implement it in another branch before proposing it to the QAM which is going to be evaluated by the QAM and the rest of the QAE before the solution is actually implemented to the system.

Chapter 3

Documentation

This document identifies the standards, practices, conventions and metrics to be used to perform activities related to the AnonCare Project. All documentations are found in the Domain Model of the repository of this project. Document standards and metrics are described in Chapter 4 of this SQAP.

Chapter 4

Standards, Practices, Conventions, And Metrics

4.1 Documentation Standards

During this project many different documents will be made. Every document has to be approved by:

- The author(s)
- The leader of the responsible team
- A member of the SQA team

In case that these three people happen to be one and the same, a second member of the responsible team has to give his approval as well. Only approved documents affect the project. The documentation standards involve the following:

- All documents must be written in English.
- Requirements on review and approval as described in chapter 5.
- Procedures involving the change of documents.
- All documents must be saved with filenames according to the purpose of each document
- All documents must pass the metrics defined later in this chapter

These standards apply to all documents, to electronic versions as well as printed ones. However the layout requirements do not apply to documents other than the project and product documents. All documents are made available through the document repository. In case of unavailability of the document repository, the Project Manager sees to it that there are three copies available of every document (latest version with the highest status of approval) in the groups workspace. The three copies consist of one copy on paper and two digital copies.

4.2 Design Standards

The design standards in the Architectural Design (AD) and Detailed Design (DD) phase will be defined or referenced in the Architectural Design Document [ADD] and Detailed Design Document [DDD] respectively.

4.3 Coding Standards

Python

- The "Zen of Python" is used as guidelines for coding standards. In addition to those guidelines team members should also include the seven ways of writing a beautiful code:
    - Return from if statements as quickly as possible.
    - Don't use an if statement when all you simply want to do is return the boolean from the condition of the if.
    - Please use whitespace it's free!
    - Don't have useless comments.
    - Don't leave code that has been commented out in the source file, delete it.
    - Don't have overly long lines.
    - Don't have too many lines within a function/method.

- Naming Conventions
    - All function names must be in lowercase
    - Function names containing multiple words are separated with the '_' sign.
    - For CRUD methods:
        - For Create, functions should start with the word 'store' followed by the type of data created (i.e store_<type_of_data>).
        - For Read, functions should start with the word 'show' followed by the type of data to be shown(i.e show_<type_of_data>).
        - For Update, functions should start with the word 'update' followed by the type of data to be shown(i.e update_<type_of_data>).

Postgresql

- Tables, functions, and triggers are made in separate files with filenames corresponding to the names mentioned
- All queries are typed in lowercase
- Naming Conventions
    - For CRUD methods:
        - For Create, functions should start with the word 'store' followed by the type of data created (i.e store_<type_of_data>).
        - For Read, functions should start with the word 'show' followed by the type of data to be shown(i.e show_<type_of_data>).

        - For Update, functions should start with the word ‘update’ followed by the type of data to be shown(i.e update_<type_of_data>).

4.4 Comments Standards

Commenting Standards are as follows:

Postgresql

- The source of codes from the internet are commented in the function where it is implemented
- Commenting of functions is not allowed
- Method type of the postgresql function is commented above the function.

4.5 Testing Standards

Python

- Gherkins is used for testing in Python
- Each feature has its own gherkin files and it contains all of its methods and scenarios
- All steps applications would be put in only file named steps.py
- Each feature implemented at steps.py is grouped; a comment before each is shown to identify the feature

4.6 Metrics

All metrics for documentations, testings, and codings are described in Chapter 5.

4.7 Compliance Monitoring

The team members will meet weekly to review that each team members followed the proposed the standards written above. Each member will present the changes he/she had made in the past week and the other members will judge the convention used by the member presenting his/her changes. Rules of compliance are as follows:

- Commits should be pushed to the repository at most 24 hours after being committed
- At the of the day/week/or 2 or 3 days, other branches should be merged back to the master branch

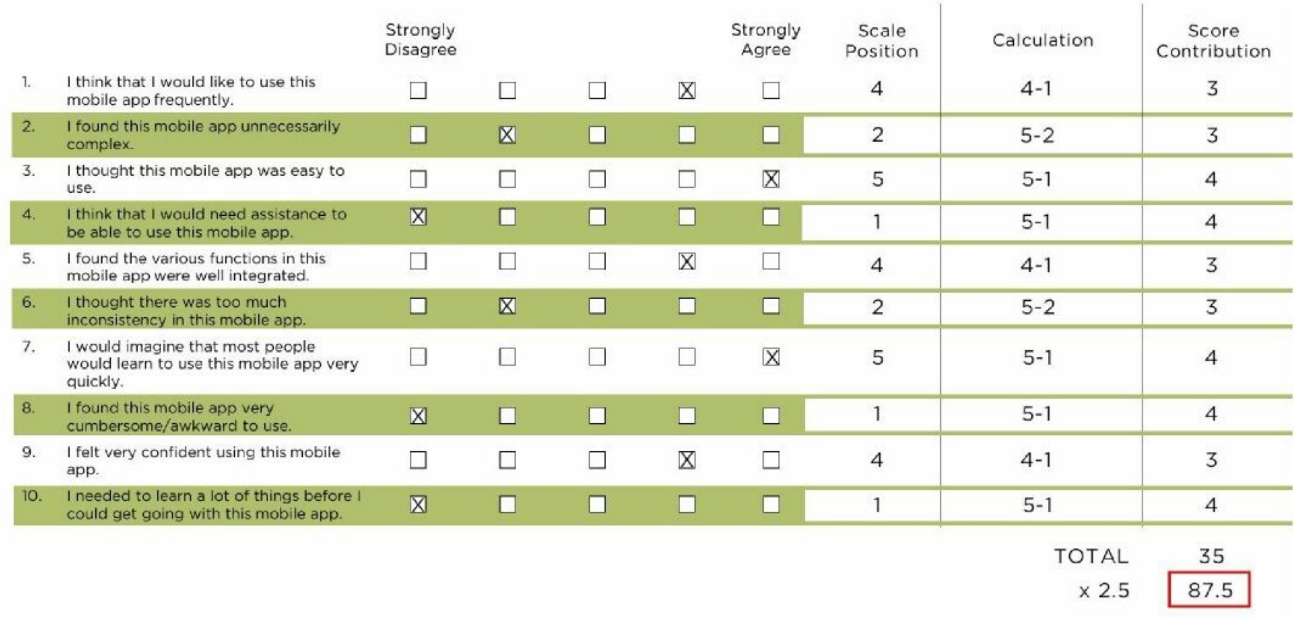
Chapter 5

Review

The SQA team will check all the project and product documents to determine whether the requirements have been met for the deliverable, ensure that products adhere to the document standards and that all SQA team members do their job properly, ensure that the SQA plan has been adhered to, verify the performance of the software and ensure that lettuce test written in Gherkins format and unit testing is carried out. Metrics for documentations, testing, coding standards and also the usability test will be shown in this chapter.

5.1 Usability Test

For usability testing, the SUS Usability test shall be used



		Strongly Disagree				Strongly Agree	Scale Position	Calculation	Score Contribution
1.	I think that I would like to use this mobile app frequently.	☐	☐	☐	☒	☐	4	4-1	3
2.	I found this mobile app unnecessarily complex.	☐	☒	☐	☐	☐	2	5-2	3
3.	I thought this mobile app was easy to use.	☐	☐	☐	☐	☒	5	5-1	4
4.	I think that I would need assistance to be able to use this mobile app.	☒	☐	☐	☐	☐	1	5-1	4
5.	I found the various functions in this mobile app were well integrated.	☐	☐	☐	☒	☐	4	4-1	3
6.	I thought there was too much inconsistency in this mobile app.	☐	☒	☐	☐	☐	2	5-2	3
7.	I would imagine that most people would learn to use this mobile app very quickly.	☐	☐	☐	☐	☒	5	5-1	4
8.	I found this mobile app very cumbersome/awkward to use.	☒	☐	☐	☐	☐	1	5-1	4
9.	I felt very confident using this mobile app.	☐	☐	☐	☒	☐	4	4-1	3
10.	I needed to learn a lot of things before I could get going with this mobile app.	☒	☐	☐	☐	☐	1	5-1	4
								TOTAL	35
								x 2.5	87.5

5.2 Documentation

Review of documentations are based on whether documentations are free from errors mentioned in Chapter 7.

5.3 Coding Standards

Review of coding standards are based on whether the codes are free from errors mentioned in Chapter 7 along with the compliance of coding standards mentioned in Chapter 4.

Chapter 6

Test

Procedure for the development process follows BDD. Each member of the development team is assigned different features for testing. The testing framework used in this development is Lettuce version 0.2.21. Test cases, on the other hand, are written in Gherkins format.

Chapter 7

Problem Reporting

When a problem is detected, it should be solved immediately. These are the different type of problems:

7.1 Documentation

- Non compliance with other project documents.
- Incompleteness
- Errors
- Grammatical Mistakes
- Sentences that are not concisely written

7.2 Code Problem

- Lack of functionality
- Wrong functionality
- Non compliance with pep8 standard
- Non compliance with team's agreed naming convention

7.3 Problem Reporting Procedure

When a problem is detected, the person who detected the problem is responsible for reporting the problem to the PM. In the case where the PM is absent, the problem is then reported to the SM, who will later report it to the PM once the PM is around.

When a problem is discovered during the meeting, the PM decides who is in-charge of the problem. Again, in the case where the PM is absent, the SM decides who will be in-charge of the problem.

7.4 Problem Solving Procedure

- The person who finds the problem first reports it to the PM.
- When the problem is solved, the PM or the SM is informed and the team checks whether the changes made solve the problem.
- If after 4 working days and the problem has not yet been solved, the PM or the SM is to call for a meeting with the group. The group then decides on how to deal with the problem in a deeper perspective.

Chapter 8

Tools Techniques And Methods

The SQA team has to make sure that appropriate tools, techniques and methods are used. The table below shows the tools, techniques and methods used during the development of the project.

Tool	**Use**
Flask	Development framework
Pycharm 2016.1	Develop source code.
Lettuce version 0.2.21	Develop and execute tests.
Visual Paradigm 13.0	Prepare data flow diagrams, use case diagram.
moqups.com	Prepare mockup ui and wireframes.
creately.com	Prepare pert chart, architectural design, and storyboard.
Google Docs	Prepare written documents.
Photoshop	Customize images
Postman	Manual Execution of tests for API

The SQA team checks their use by means of random checks.

With respect to the tool used during this project special interest is paid to:

- Availability of the tools. (Has every group member access to the tools?).
- Knowledge. The group members working with the tools must have the necessary skills to work with the tools (see also chapter 13, Training).
- The tools must work properly. (Are there errors or malfunctions in tools? Is there enough capacity?).

Every used tool will be checked at least once before use and once during use. When problems appear the SQA decides together with the PM and CM if the problem can be solved, or if the tool must be replaced by an alternative.

Chapter 9

Code Control

It is the senior manager's responsibility to assure the correct handling of the code.

The following has to be valid:

- Documents are available to all people who are authorized to access them and to no one else.
- All versions of a document are available.
- Name conventions are consequently used.

The project manager checks if the procedures and standards are handled properly. This is done by reviews and random checks. Problems are reported to the project manager.

Chapter 10

Media Control

The project manager checks if the procedures and standards are handled properly. This is done by reviews and random checks. Problems are reported to the project manager.

Chapter 11

Supplier Control

The MSU-IIT Clinic supplies the development team with necessary data from their stored medical records of students, faculties and staffs. Data provided to us from the MSU-IIT Clinic were list of most common illness diagnosed during the previous school year.

Chapter 12

Records collection, maintenance and retention

All document sources and source code will be available in the Git repository. Document will be made available through Git. The directory structure for the team directory is as follows:

Boundary Model/	Contain the AHS Story
Code Model/	Contains codes
Domain Model/	Contain the Charter
Internals Model/	Contains the Data flow diagram, ERD, and Use Cases

The structure for the project code :

Code Model/app	Source code, api, stored procedure
Code Model/test	Lettuce testing

Chapter 13

Training

During the project, every member of the group should have the skills needed to develop the project. Each member is given at most 2 weeks to learn all the skills needed for the project.The PM and SM will assess the level of knowledge for the task in the group and then they decide whether special action needs to be taken. The following table shows the skills needed for the project and its corresponding references.

Skill Needed	References
Flask	www.**tutorialspoint**.com/**flask**
JavaScript	www.**tutorialspoint**.com/**javascript**
Lettuce	lettuce.it/
HTML	www.**tutorialspoint**.com/**html/**
CSS	www.**tutorialspoint**.com/**css/**
Jasmine	● http://jasmine.github.io/2.1/introduction.html ● http://evanhahn.com/how-do-i-jasmine/

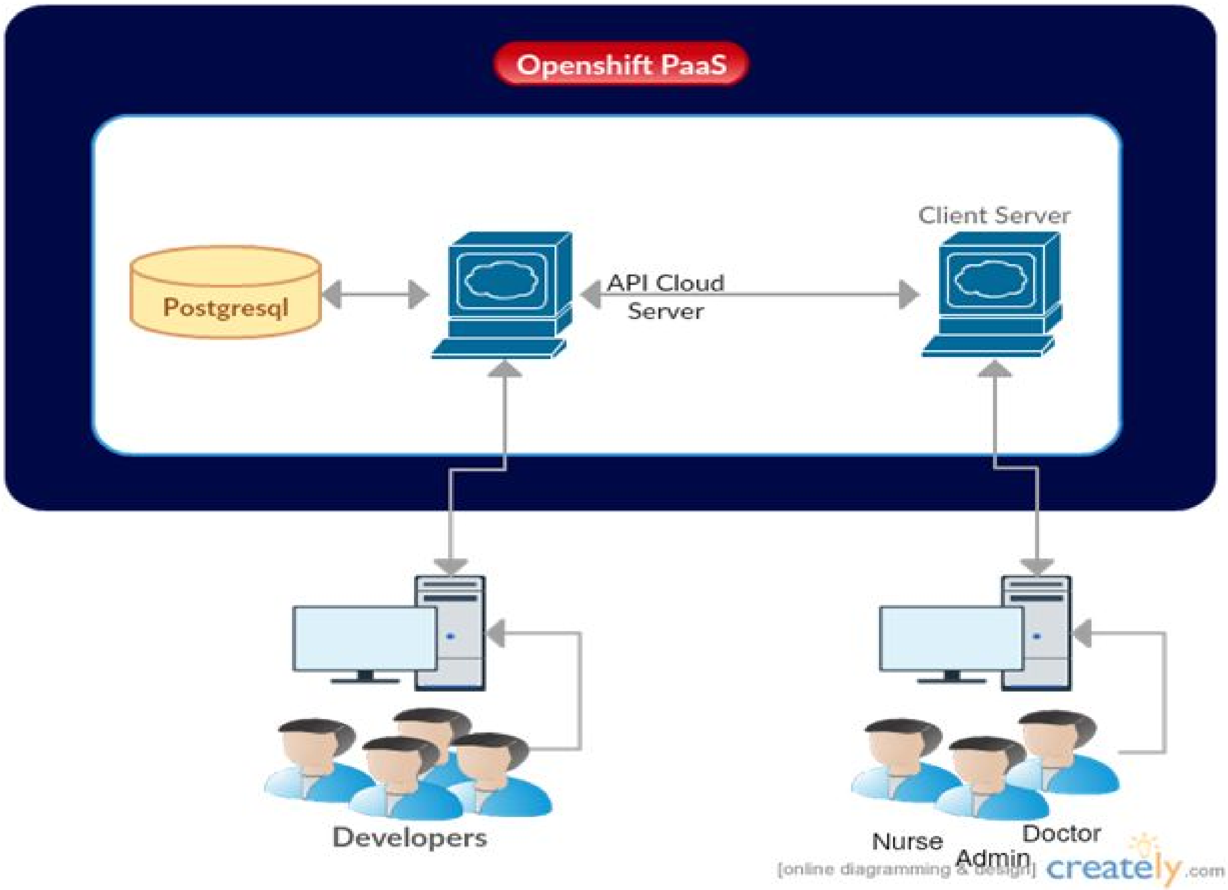
Chapter 14

Risk management

In the SPMP[4], the risks of the project are described. During progress meetings the occurrence of any of the risks described must be discussed and the PM must see to it that the necessary course of action is taken. The SM will assist him/her in this task.

APPENDIX

Software Architecture Diagram

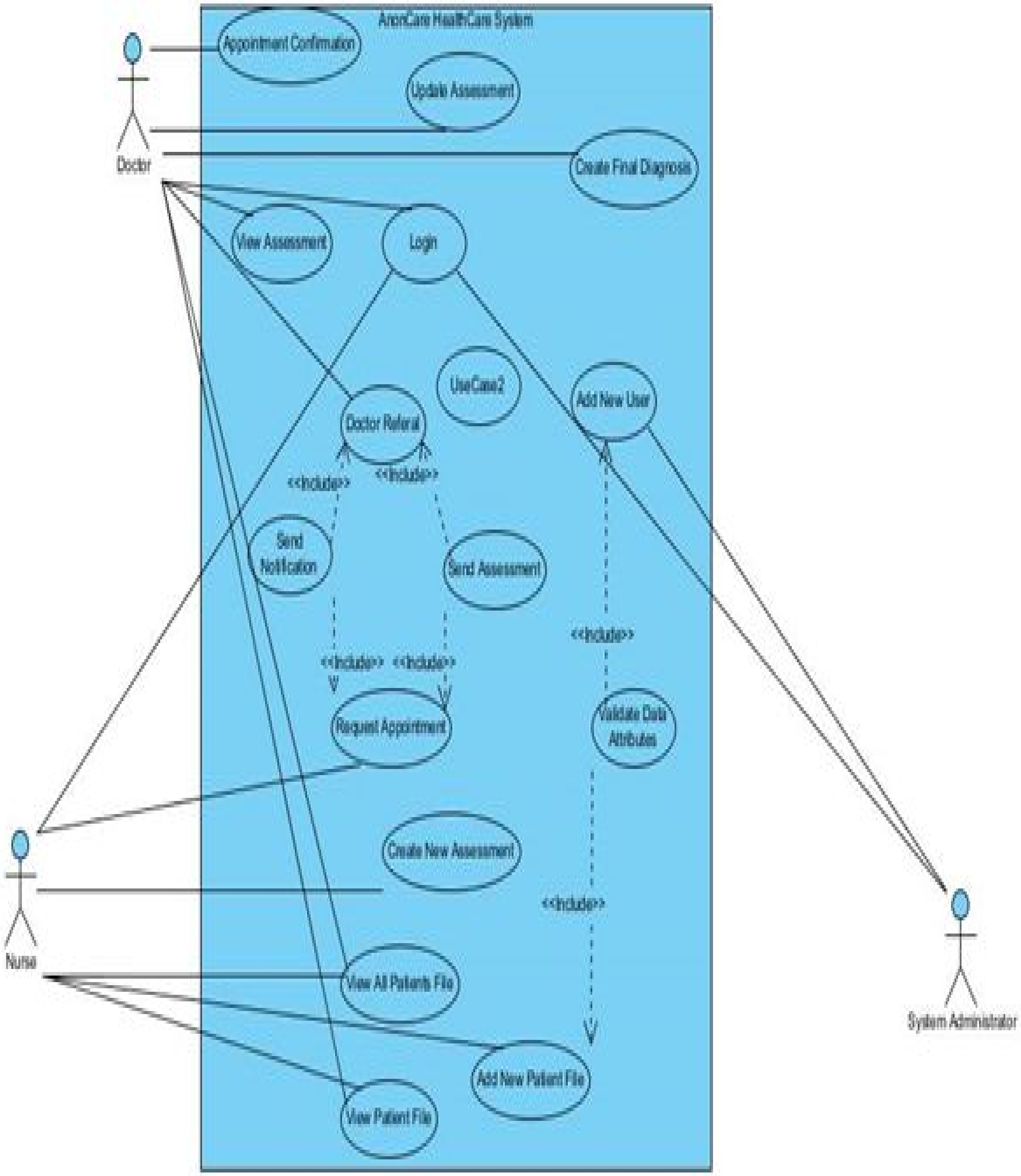


In using OpenShift PAAS, we used two gears. One gear is for our API Cloud Server. This API Cloud Server contains Postgresql, Python. The other gear is for the Client Server which contains PHP 5.4. The user, either the Nurse, System Administrator and Doctor, directly interacts with the User Interface(UI) of the system and the UI takes the inputs and sends it to the Client Server. The User Interface and the Client Server is a two-way interaction. Once the Client Server has accepted the user's inputs, it then sends it to the API Cloud Server. The API Cloud Server is the brain of the system. The interaction between the API Cloud Server and the Client Server is a two-way one. Once the API Cloud Server verifies the request from the Client Server, the API Cloud Server then sends it to the Postgesql Database. The API Cloud Server and the Postgresql Database is also a two-way interaction. Once the Postgresql has finished doing its query, it returns a response to the API Cloud Server, and then the API Cloud Server returns a JSON data to Client Server and then the Client Server implements the action to be done in the UI.

Usability Test Conclusion

Based on our System Usability Scale results, it was apparent that the nurses and the doctors liked the system. And they really thought that they would really be able to use the system very well. However, changes to the UI are needed to be made. The nurses find it rather unnecessarily complex and would prefer a different style as to how to display the forms they need.

Class Diagram


AronCare HealthCare System
Appointment Confirmation
Update Assessment
Create Final Diagnosis
Doctor
View Assessment
Login
UseCase2
Add New User
Doctor Referal
<<Include>>
<<Include>>
Send Notification
Send Assessment
<<Include>>
<<Include>>
<<Include>>
Request Appointment
Validate Data Attributes
Create New Assessment
<<Include>>
Nurse
View All Patients File
Add New Patient File
View Patient File
System Administrator