Analysis Specification Report

On

E-Gig Nepal

Rohit Sai

00162939

Computing Project

Level 5 Diploma in Computing

Softwarica College of IT & E- Commerce

Kathmandu, Nepal

Table of Contents

Table of Figures:

2.1. Introduction:

Analysis is the second phase of the system development lifecycle. This phase defines the requirements of the system and also how the stated requirements will be accomplished. In this phase, the problem that the client is facing and the reason why the system is needed to be developed will be stated and tried to be solved. The result that will be delivered will be presented at the end of this phase of the requirement documentation. This documentation clearly describes what is to be built or what need to be built. Since this phase describes what is to be built, so It is also known as "**WHAT**" phase. The documentation tries to describe all the pre-requisite from the client's view by describing the build area and all the connections of the application implementation. In this phase, we all understand what the customer / client want. The main purpose of this phase is to gather various information from an organisation, process it, summarise it, document it and produce a structured specification for a new/upgraded system.

(Infolab.stanford.edu, 2017)

2.2. An Overview of E-Gig Nepal:

E-Gig Nepal is a web application that will be developed to make the searching of gigs or shows of bands in Nepal easy. This application supports promoting the gigs / band shows. New or old band's information is also posted in this application to help the audiences know about their favorite band. The organizers / band members are provided with a separate account to join the application so that they can post their upcoming events or shows and help get promoted for free. Also, admin can post numerous posts related to bands and happening shows.

2.3. Requirements:

The requirements are the description of all aspects of the current information system (if exists) or any kind of problems that exists for which the system is going to be developed. Also, this includes the reference on how to solve the problems or improve or change the current system. A complete description (answer) for a new/upgraded system is also included in the requirements. The requirement analysis should be able to answer the following questions:

- What is being done?
- How is it being done?
- Who is doing it?
- When is it being done?
- Why is it being done?
- Can it be improved and if so, how?

A software requirements specification (SRS) is also a part of this phase, which contains details of the functional requirements of the system, such as the interactions between the users and the software and the non-functional requirements, e.g. the quality standards, etc.

2.3.1. Functional Requirements:

Admin

ID	Title	Description	Rational	Dependencies
FR1	Admin Login	The admin can log in by entering the password and username on admin panel.	Admin authentication.	-
FR2	Add Band Profile	The admin of the system can add the band profiles and details.	Add details about Bands	FR1
FR3	Edit Band Profile	The admin of the system can edit pre-saved or posted profile of the band.	Change the details of the post of bands details.	FR1, FR2
FR4	Read Event Promotion Messages	The admin can read the messages sent by the organizers or the band members about promoting their event.	View the messages from organizers of the event.	FR1
FR5	Add Event	The admin can add the events that the organizers has requested to post.	Add details about the upcoming events.	FR1, FR4
FR6	Edit Events	The admin can edit the pre-posted events if any mistake has occurred.	To edit the pre-posted events.	FR1, FR4, FR5
FR6	Add News related to conducted events	The admin can post the news about the event concluded and the details on how the event went.	To add news of the events.	FR1

Organizer / Band Members:

ID	Title	Description	Rational	Dependencies
FR1	Registration	The organizers / band members can register for their account.	User Registration.	-
FR2	Login	The organizers / band members can login their account by their unique id and password.	User (Organizer/ Band Members) Authentication.	FR1
FR3	View Profile	The organizers / band members can view their details that were stored during account registration process.	To see the profile of themselves.	F1, F2
FR4	Edit profile	The organizers / band members can edit their profile by replacing their old details.	To modify their own personal details.	FR1, FR2, FR3
FR5	View Band profiles	The organizers / band members can get information about their favorite bands.	To see the details of bands already posted.	F1, F2
FR6	View Events	The organizers / band members can view all the upcoming events to plan for their own event or join	To see already posted events details.	F1, F2
FR7	Send Event Promotion Messages	The organizers / band members can send their message to admin about the event that they will be conducting	Send prompt to admin about their event	FR1, FR2
FR8	Send Band Profiles	The organizers / band members can send the details of band they want to promote	Send Prompt to admin to add new band profile	FR1, FR2

Audience / Normal User:

ID	Title	Description	Rational	Dependencies
FR1	Registration	The normal user can register for their account	User Registration.	-
FR2	Login	The normal user can login their account by their unique id and password.	Normal User Authentication.	FR1
FR3	View Profile	The normal user can view their details that were stored during account registration process.	To see the profile of themselves.	F1, F2
FR4	Edit profile	The normal user can edit their profile by replacing their old details.	To modify their own personal details.	FR1, FR2, FR3
FR5	View Band profiles	The normal user can get information about their favorite bands.	To see the details of bands already posted.	F1, F2
FR6	View Events	The normal user can view all the upcoming events to plan for their own event or join	To see already posted events details.	F1, F2

2.3.2. Non – Functional:

ID	Title	Description	Rational	Dependencies
NF1	Efficiency	It should perform all it’s task efficiently	Maintain Efficiency	NF2, NF3, NF6, NF8
NF2	Functionality	Are all the functions of the system fulfilled or not	Maintain requirement analysis functions.	-
NF3	Accuracy	It should be performing accurate tasks	Helps maintain accuracy level	NF1, NF2, NF5
NF4	Robustness	It should be mitigating the toughness.	To minimize toughness.	NF3, NF6
NF5	Reliability	It should be fully dependable or which can be trusted on.	Maintain dependability	NF1, NF3
NF6	Ease of Use	It should be easy to use after completion.	To make use smooth.	NF5, NF3, NF4
NF7	Extendibility	It should be easier to add features or other parts during development period	To make sure new features can be added.	NF8, NF1
NF8	Maintainability	It should be maintainable if any problems occur during the time of system use.	To prevent from work, stop.	NF7
NF9	Security	The security of flow/movement of data should be maintained within server and application.	Maintain security.	NF5

2.3.3. Prioritization:

The Stakeholders have a list of long requirements. But they can't always have everything stated in the requirement as they want because we can't give everything they want. So, for this requirement needs to be prioritized. This is not because we don't like them or we have any problems with them but because most projects have a limited budget and time to complete them. So, we ensure the focus of most important functionalities in the given requirements. (Famuyide and Famuyide, 2017)

So, for prioritizing the requirements we can use MoSCoW technique.

This technique has four different focus area divided as follows:

MUST(M):

This means that the requirements having MUST category has to be fulfilled anyhow before the submission of the project.

SHOULD(S):

This means that the requirements having SHOULD category has to be fulfilled if possible before the submission of the project.

COULD(C):

This means that the requirements having COULD category is nice to be included if possible. The project without COULD functions is also acceptable for the final submission.

WON'T or WOULD(W):

This means that the requirements having WON'T or WOULD category are the functions that stakeholders want to have, but agreed not to implement in the current version of the system due to various factors.

The prioritization of the requirements is listed below:

Functional Requirements

Admin

ID	Functional Requirements	MoSCow
FR1	Admin Login	Must Have
FR2	Add Band Profile	Must Have
FR3	Edit Band Profile	Should Have
FR4	Read Event Promotion Message	Should Have
FR5	Add Event	Must Have
FR6	Edit Events	Should Have
FR7	Add News related to conducted events	Must Have

Organizer /Band Members

ID	Functional Requirements	MoSCow
FR1	Registration	Must Have
FR2	Login	Must Have
FR3	View Profile	Should Have
FR4	Edit Profile	Should Have
FR5	View Band profiles	Must Have
FR6	View Events	Should Have
FR7	Send Event Promotion Messages	Must Have
FR8	Send Band Profiles	Should Have

Audience/Normal User

ID	Functional Requirements	MoSCow
FR1	Registration	Must Have
FR2	Login	Must Have
FR3	View Profile	Should Have
FR4	Edit Profile	Should Have
FR5	View Band profiles	Must Have
FR6	View Events	Should Have

Non-Functional Requirements

ID	Non-Functional Requirements	MoSCow
NF1	Efficiency	Must Have
NF2	Functionality	Must Have
NF3	Accuracy	Must Have
NF4	Robustness	Should Have
NF5	Reliability	Must Have
NF6	Ease of Use	Should Have
NF7	Extendibility	Should Have
NF8	Maintainability	Should Have
NF9	Security	Must Have

2.4. Use Case:

Use Case is the model which shows the functionalities that different users of the system can perform. There are various agents which makes changes to the system or interact with system. All these agents which interacts with the system is known as actor. All the actors / users of the system are identified in the use case and their interaction with the system is also presented. So, use case can be defined as a model which identifies all the agents interacting with the system and present their relationship with the system. (tutorialspoint.com, 2017)

The purpose of the use case is as follows:

- Used to gather requirements of a system.
- Used to get an outside view of a system.
- Identify external and internal factors influencing the system.
- Show the interacting among the requirements are actors.

The figures below show the Use Case for the E-Gig Nepal:

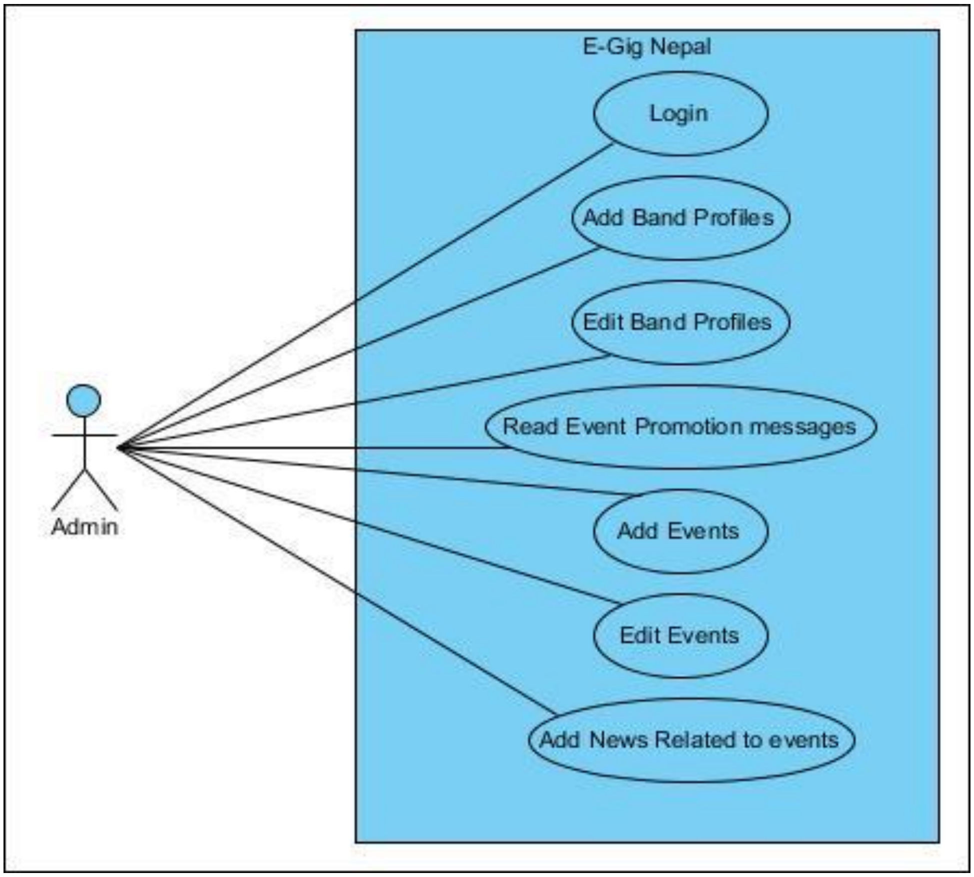

Figure 1 Use Case of Admin

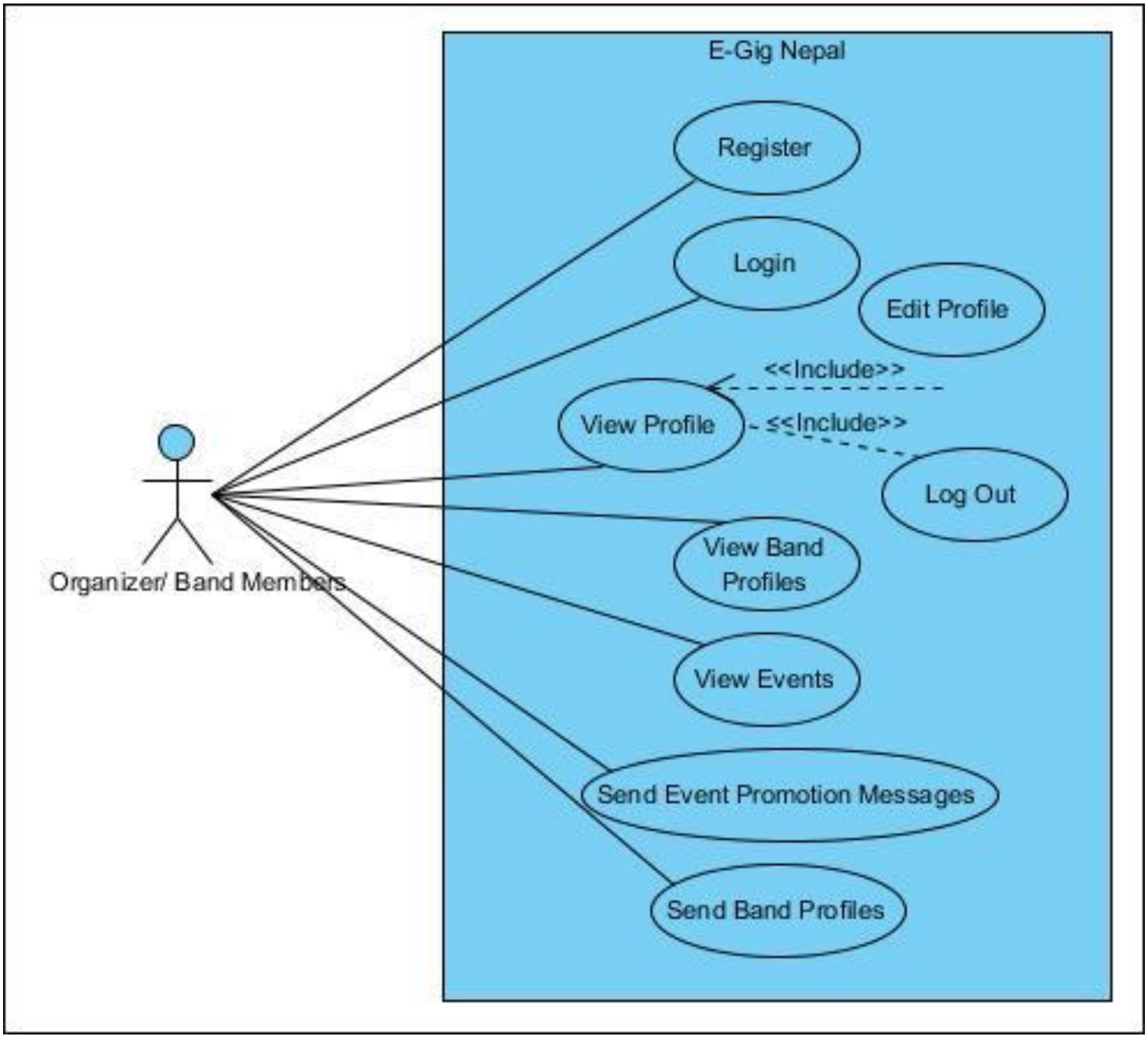

Figure 2 Use Case for Organizer/ Band Members

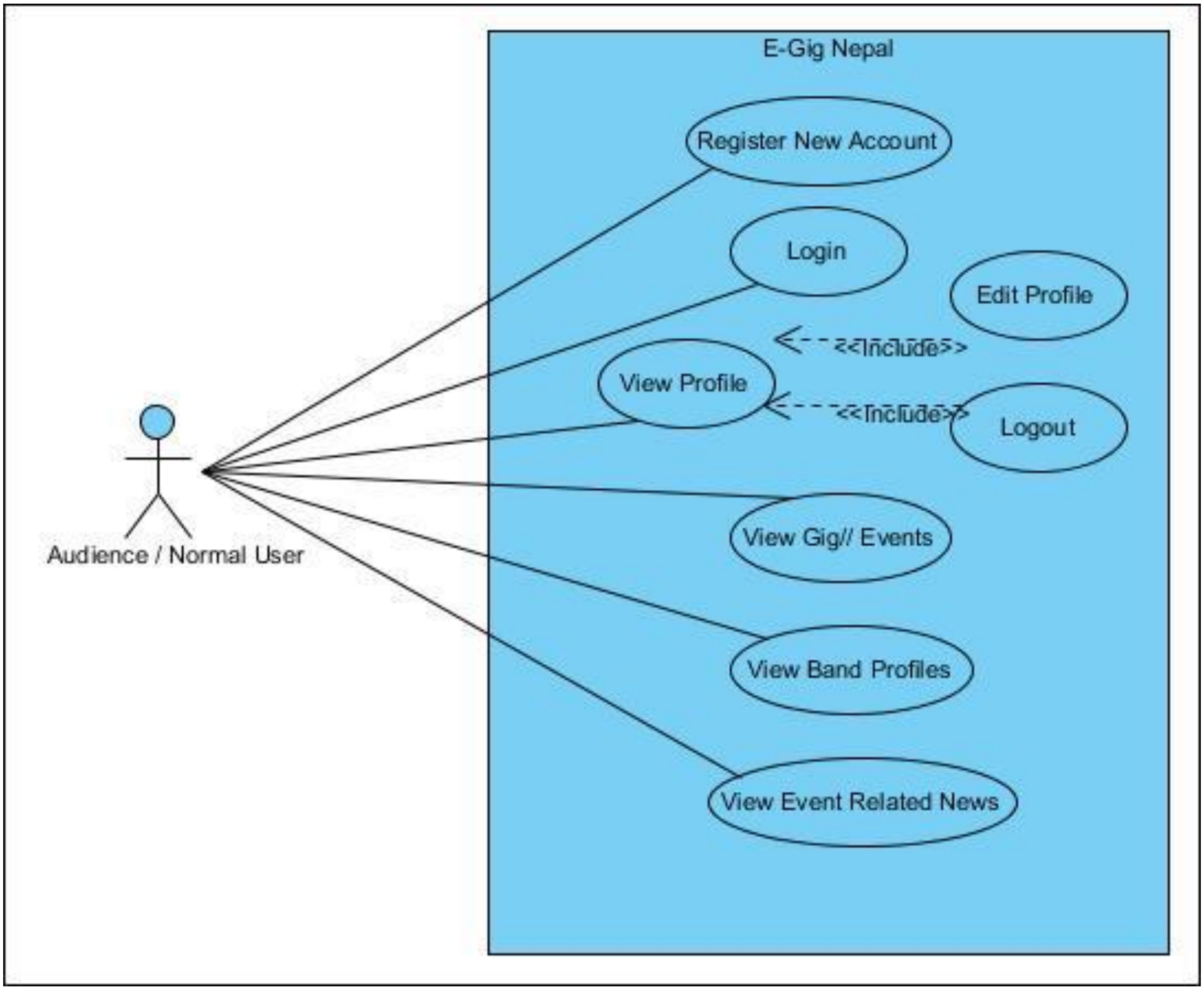

Figure 3 Use Case for Normal User/ Audience

2.5. System Architecture:

Systems Architecture is a basic discipline to handle objects (existing or to be created) called "systems", in a way that supports reasoning about the structural properties of these objects. It is a reaction to the applied and reasonable challenges of the explanation and the plan of complex systems. It determines the system's behavior and dynamic. For this, I decided to use MVC (Model View Controller) design pattern and PHP for developing application and MySQL as per database. (Lix.polytechnique.fr, 2017)

The figure below (Onextrapixel.com, 2017) shows the MVC Pattern of system architecture for Web Application developed using PHP:

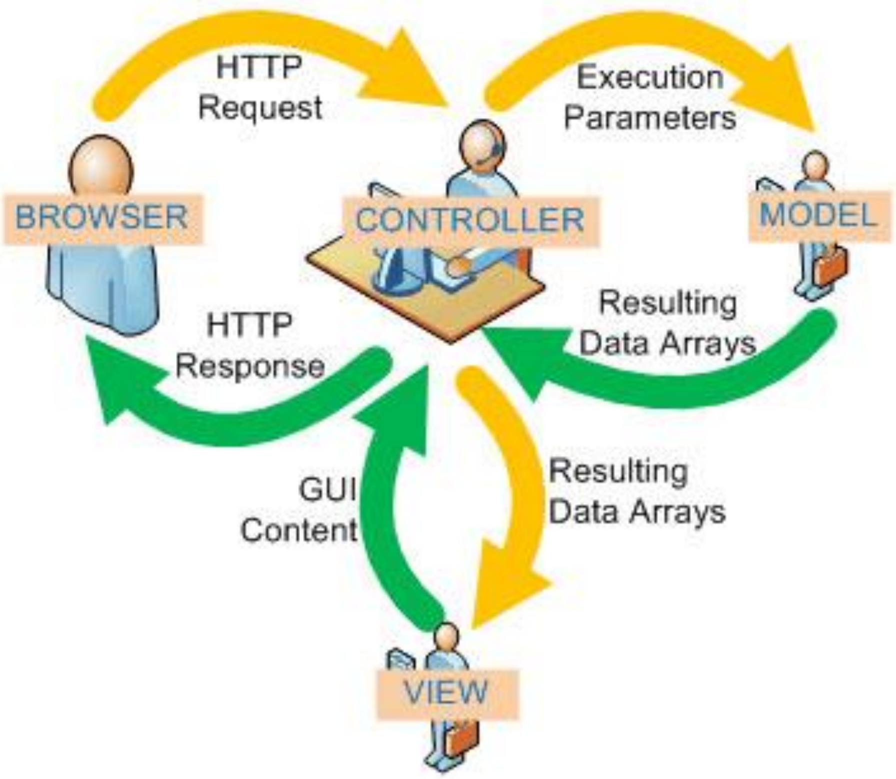

Figure 4 System Architecture showing MVC

2.6. Initial Class diagram:

We use Natural Language analysis to find out the class name, attributes and the variables. We use NLA process to find it. In this process, we list out all the Nouns, Adjectives and verbs from the given scenario. We then separate the noun as potential classes, adjectives as potential attributes and verbs as potential functionality. We then filter all the listed nouns, adjectives and verbs if there are duplicate entries and if are out of scopes.

Class diagram is the diagram that explains all the classes with their attributes, functionality and relations between other classes. Class diagrams are difficult to construct so we use Candidate list and the draft class diagrams during the process of construction of class diagram. (tutorialspoint.com,2017)

List of Nouns, Adjectives and Verbs from the scenario:

Noun	Adjective	Verb
Gigs, Musical Shows, fans, details, organizers, promoters, event planners, details, taste, information, genres, background information, venue, date, list, admin, promotion, Contact, profiles, tickets, web application, Bands	Developed, interested, manually, registered, valid, successful, edit, genuine	promoted, Viewed, find, pay, post, interested, searched, found, search, E-mail, showed, viewed, recorded,

Initial list of possible classes and actions:

Classes	Actions
Gigs, Fans, Organizers, Admin, Band Profile	Find Gigs, Record Gigs, Record Fans, Record Organizers, Record Band Profile, Find Band Profile

The First Draft and the Second Draft Initial - Class Diagrams are prepared by following the NLA process at first. The diagrams are below. In case of the unclarity of the image in the documentation, the diagrams have been attached along with the documentation inside documentation folder.

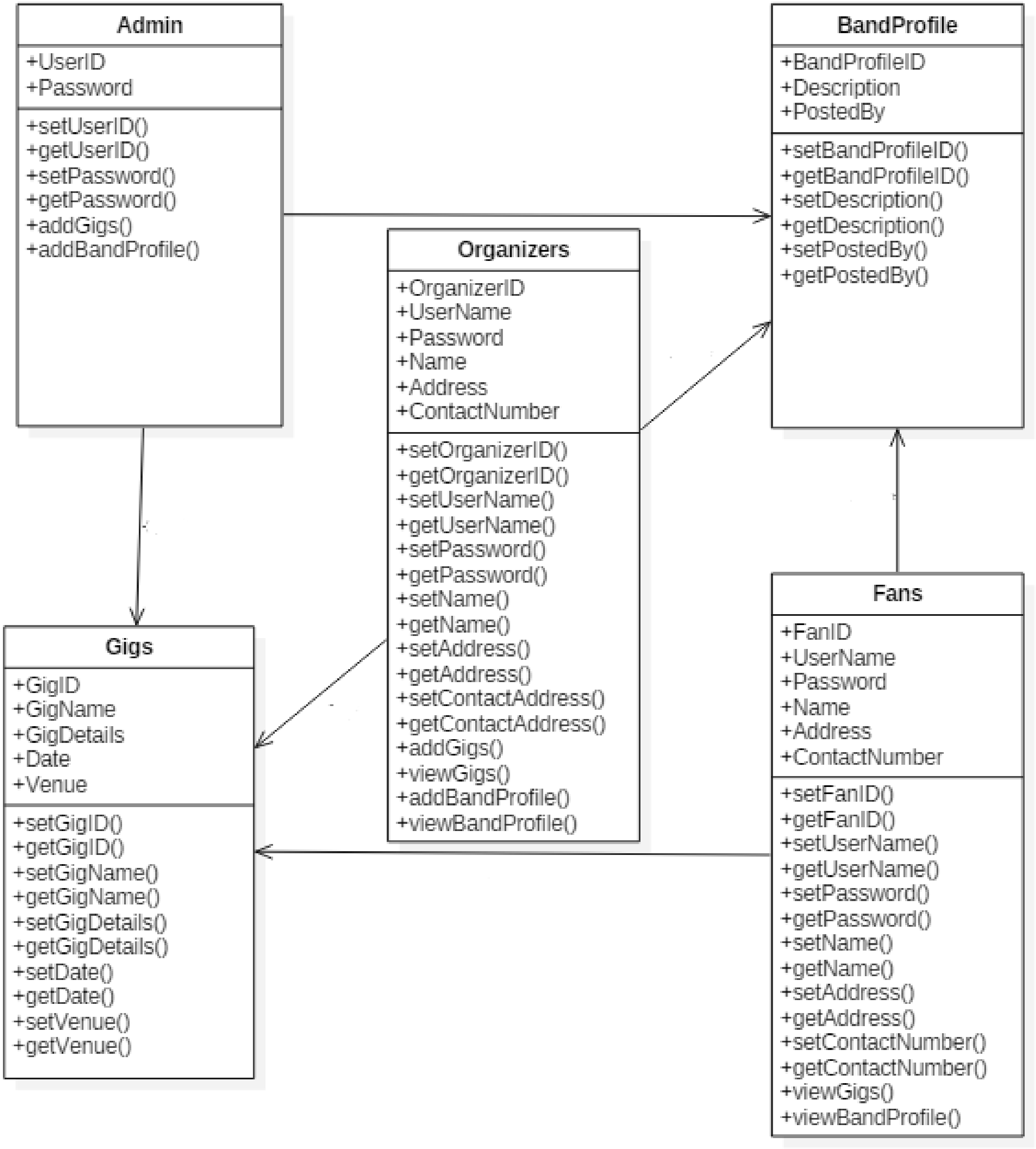

Figure 5 First Draft Class Diagram

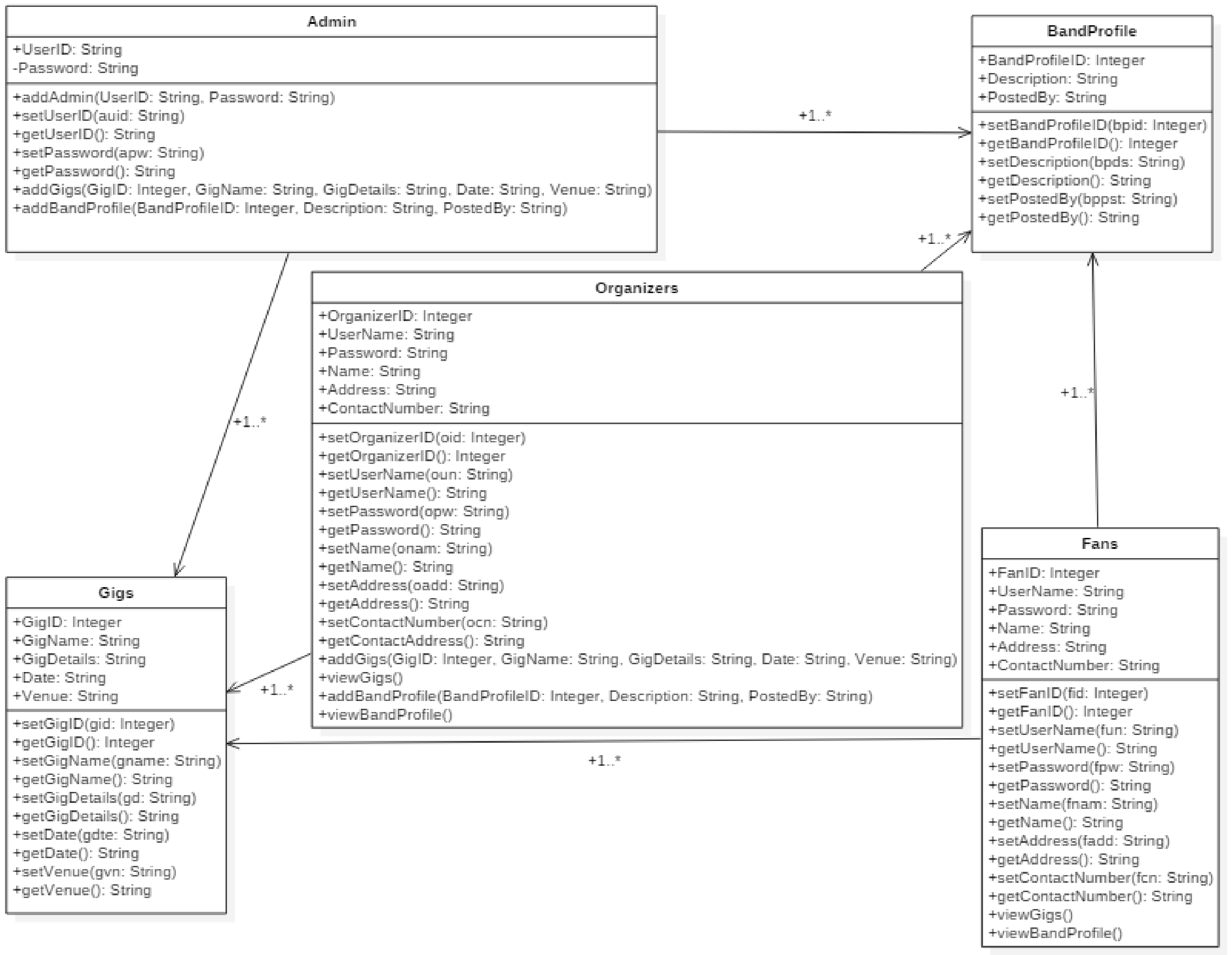

Figure 6 Second / Final Draft Class Diagram

2.7. Rich Picture:

Rich Picture is the pictorial explanation of the existing system having a problem or a new system which needs to be made. This picture shows the view of whole system and can enable better planning and understanding of a system. It is created consulting the stakeholders of the system and is usually drawn by hand and include structure, processes, issues or development.

The figure below shows the rich picture of E-Gig Nepal.

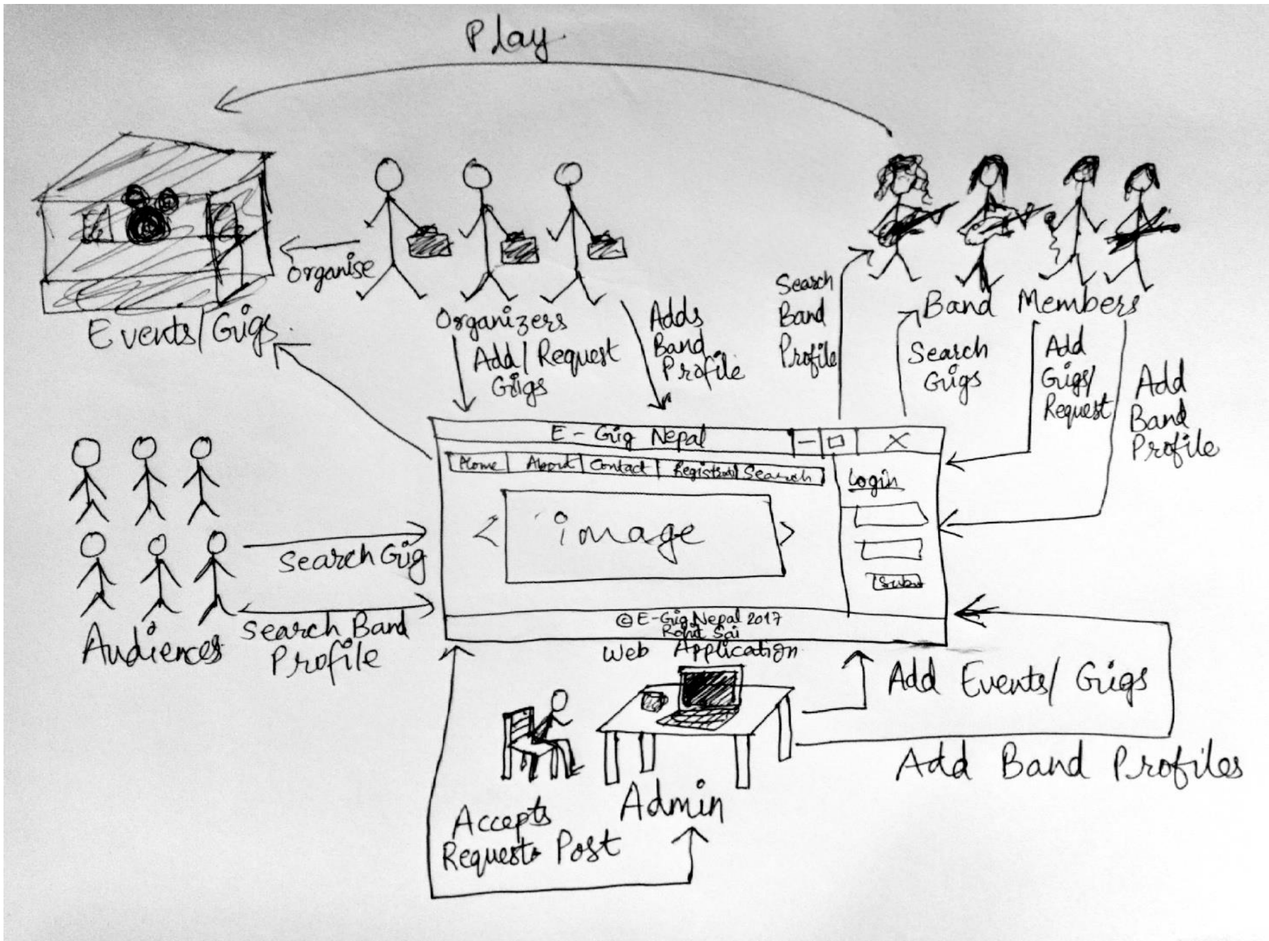

Figure 7 Rich Picture showing E-Gig Nepal's System

Conclusion:

Finally, the second phase of the project i.e. analysis phase has been concluded successfully. In this documentation, we have listed all the Software Requirement Specification (SRS) having both functional and non-functional requirements. All these requirements have been prioritized using MoSCoW technique. Also 3 different use case diagrams have been made for three different kinds of the users on the system. The interactions between the user and the system is showed there. The system architecture has been shown and MVC design pattern is explained. We have also made an initial class diagram initially following NLA process. At last, we have also made Rich Picture to make the requirements clear and describe the complete system. Now we can conclude this phase and continue for design phase of system development.

References:

Infolab.stanford.edu. (2017). *The Analysis Phase*. [online] Available at: http://infolab.stanford.edu/~burback/watersluice/node4.html [Accessed 11 Apr. 2017].

Famuyide, S. and Famuyide, S. (2017). *MoSCoW : Requirements Prioritization Technique*. [online] Business Analyst Learnings. Available at: https://businessanalystlearnings.com/ba-techniques/2013/3/5/moscow-technique-requirements-prioritization [Accessed 13 Apr. 2017].

www.tutorialspoint.com. (2017). *UML - Use Case Diagrams*. [online] Available at: https://www.tutorialspoint.com/uml/uml_use_case_diagram.htm [Accessed 17 Apr. 2017].

Lix.polytechnique.fr. (2017). *What is Systems Architecture ?*. [online] Available at: https://www.lix.polytechnique.fr/~golden/systems_architecture.html [Accessed 20 Apr. 2017].

Onextrapixel.com. (2017). *Cite a Website - Cite This For Me*. [online] Available at: http://www.onextrapixel.com/wp-content/uploads/2010/05/mvc.jpg [Accessed 20 Apr. 2017].

www.tutorialspoint.com. (2017). *UML - Class Diagram*. [online] https://www.tutorialspoint.com/uml/uml_class_diagram.htm [Accessed 21 Apr. 2017].