



thydash

Full Documentation

thydash: an eTourism Platform using live video streaming from Android Based Devices.

Acknowledgement

We have completed the implementation part of thydash successfully. The word “we” includes not only the members of the team, but also some other individuals who should assume our gratitude. Mr. Kifle and Mr. Tinsae, our advisors, for advising us throughout the whole process effectively and efficiently; Mr. Endale, lecturer in ASTU, for his comments in designing thydash; and developers of Django web framework, for building the open source and batteries included python web framework are some to mention. Let all you have our heartfelt thanks.

Table of Contents

ACKNOWLEDGEMENT	I
TABLE OF CONTENTS	II
LIST OF TABLES	VIII
ACRONYMS	XIV
ABSTRACT	XV
CHAPTER 1: INTRODUCTION.....	1
1.1 BACKGROUND	1
1.2 STATEMENT OF THE PROBLEM	2
1.3 PURPOSE OF THE SYSTEM	3
1.4 TEAM COMPOSITION	3
1.5 OBJECTIVES OF THE PROJECT	3
1.5.1 General Objectives.....	3
1.5.2 Specific Objectives	4
1.6 FEASIBILITY	4
1.6.1 Technical Feasibility	4
1.6.2 Operational Feasibility	4
1.6.3 Economic Feasibility	5
1.6.4 Schedule Feasibility	5
1.7 SCOPE AND LIMITATION	5
1.7.1 Scope.....	5
1.7.2 Limitations.....	6
1.8 SIGNIFICANCE OF THE PROJECT	6
1.9 METHODOLOGY	6
1.10 DEVELOPMENT TOOLS.....	7

1.11 TESTING PROCEDURES	7
1.11.1 Unit Testing.....	8
1.11.2 Integration Testing.....	9
1.11.3 System Testing.....	9
1.12 OVERVIEW OF PROJECT PHASES	11
1.13 TASK AND SCHEDULE.....	12
CHAPTER 2: LITERATURE REVIEW	13
2.1 INFORMATION TECHNOLOGY IN TOURISM SECTOR.....	13
2.2 LIVE VIDEO STREAMING FROM ANDROID DEVICES	14
2.2.1 Overview.....	14
2.2.2 Live Video Streaming	14
2.2.2.1 Types of Live Video Streaming	14
2.2.2.2 Android OS	16
2.2.2.3 (En De) Coding	17
2.2.2.4 Codec Formats	17
2.2.2.5 Internet Protocols	18
2.2.2.6 RTP	18
2.2.2.7 RTSP.....	19
2.2.2.8 Flash Player.....	19
2.2.3 Live Streaming Providers	19
2.2.3.1 YouTube Live	19
2.2.3.2 Twitch	20
2.2.3.3 Periscope	20

CHAPTER 3: PROPOSED SYSTEM	21
3.1 OVERVIEW	21
3.2 FUNCTIONAL REQUIREMENTS	21
3.3 NON FUNCTIONAL REQUIREMENTS	23
3.4 SYSTEM MODEL	24
3.4.1 UseCase Model	24
3.4.2 UseCase Description.....	27
3.5 OBJECT MODEL	38
3.5.1 Data Dictionary	38
3.5.2 Class Diagram.....	41
3.5.3 Object Diagrams	42
CHAPTER 4: SYSTEM DESIGN	45
4.1 OVERVIEW	45
4.2 DESIGN GOALS	45
4.3 PROPOSED SYSTEM ARCHITECTURE	46
4.3.1 High Level Design	46
4.3.2 Subsystem Decomposition.....	48
4.3.3 Hardware / Software Mapping.....	49
4.3.4 Persistent Data Management.....	50
4.3.5 Component Diagram	50
4.3.6 Deployment Diagram	51
4.3.7 Boundary Conditions	52
4.3.8 Database Design.....	55

4.3.9 Access Controlling.....	57
4.3.10 User Interface Design	58
CHAPTER 5: FEATURES LIST	60
5.1 USER STORIES	60
5.1.1 Android User Stories	60
5.1.2 Web User Stories	60
CHAPTER 6: FEATURES IMPLEMENTATION	63
6.1 USER MANAGEMENT MODULE.....	63
6.1.1 Features.....	63
6.1.2 MVC Class Diagram.....	63
6.1.3 Screenshots	64
6.1.4 Tools and Technologies Used	64
6.1.5 Prototype Setup	64
6.2 EVENT MANAGEMENT MODULE	65
6.2.1 Features.....	65
6.2.2 MVC Class Diagram.....	65
6.2.3 Screenshots	66
6.2.4 Tools and Technologies Used	67
6.2.5 Prototype Setup	67
6.3 LIVE STREAMING MODULE	68
6.3.1 Features.....	68
6.3.2 Design Diagram	68

6.3.3 Screenshots	69
6.3.4 Tools and Technologies Used	70
6.3.5 Prototype Setup	70
6.4 INSTANT MESSAGING MODULE	70
6.4.1 Features	70
6.4.2 Design Diagram	71
6.4.3 Tools and Technologies Used	71
6.4.4 Prototype Setup	71
CHAPTER 7: TESTING	72
7.1 OVERVIEW	72
7.2 OBJECTIVE	72
7.3 SCOPE	72
7.4 RESOURCE (RELATION WITH OTHER DOCUMENTS)	73
7.5 SCHEDULE	74
7.6 FEATURES TO BE TESTED OR NOT TO BE TESTED	78
7.6.1 Features to Be Tested	78
7.6.2 Feature Not To Be Tested	78
7.7 PASS/FAIL CRITERIA	79
7.7.1 Component Pass/Fail criteria	79
7.7.2 Integration Pass/Fail criteria	79
7.7.3 System Pass/Fail criteria	79
7.8 APPROACH	79
7.8.1 Test case alignment with Test phases	80

7.9 TEST CASE SPECIFICATION	80
7.10 ESTIMATED RISK AND CONTINGENCY PLAN	98
7.11 CONCLUSION	98
CONCLUSION AND RECOMMENDATION	99
REFERENCES	100
ANNEX	101
I. Code Snippets	101
II. Algorithms and Standards.....	104
Search: Elastic Search.....	104
Video Encoding: H.264.....	105
Audio Encoding: AAC.....	105
Color Space: YUV.....	105
Instant Messaging: Socket.IO.....	106

List of Tables

Table 1: Team Composition.....	3
Table 2: Time schedule	12
Table 3: UCD for Create Account	27
Table 4: UCD for Create Event.....	28
Table 5: UCD for Join Event	30
Table 6: UCD for broadcast live event	32
Table 7: UCD for Watch Live Event	35
Table 8: UCD for download past events	36
Table 9: User Data Dictionary	38
Table 10: Event Data Dictionary	38
Table 11: Video Data Dictionary	39
Table 12: Payment Data Dictionary.....	39
Table 13: Discussion Data Dictionary	39
Table 14: Rate Data Dictionary	40
Table 15: Comment Data Dictionary	40
Table 16: Group Data Dictionary.....	40
Table 17: UCD for Start Server	52
Table 18: UCD for Stop Server	53
Table 19: UCD for Configure Server.....	53
Table 20: UCD for Install Required Packages.....	53
Table 21: UCD for Deploy Updates	54

Table 22: UCD for Configure Database	54
Table 23: UCD for Run Migration Scripts.....	54
Table 24: Thydash Access Controlling.....	57
Table 25 Unit Testing 001: Q&A module on web	74
Table 26 Unit Testing 002: Q&A module on android app.....	74
Table 27 Unit Testing 003: Sign Up with Google.....	74
Table 28 Unit Testing 004: Create Event module on web and YouTube	75
Table 29 Unit Testing 005: Searching Event	75
Table 30 Unit Testing 006: Comment Feature on Past Event.....	75
Table 31 Unit Testing 006: Streaming from Android camera	76
Table 32 Unit Testing 007: Braintree Payment module.....	76
Table 33 Integration Testing 001	76
Table 34 Integration Testing 002	77
Table 35 Integration Testing 003	77
Table 36 Integration Testing 004	77
Table 37 Integration Testing 005	78
Table 38 TC001: User Profile - Phone Number Validity	80
Table 39 TC002: User Profile- Phone Number validity	81
Table 40 TC003: Event Creation - Price Validity 0.....	81
Table 41 TC004: Event Creation - Price Validity 1.....	82
Table 42 TC005: TC005: Event Creation - Price Validity 2.....	83
Table 43 TC006: Event Creation - Event Date Validity 0	83

Table 44 TC007: Event Creation – Event Date Validity 1	84
Table 45 TC008: Event Creation - Event Date Validity 2	85
Table 46: TC009: Event Creation - Time Range Validity 0.....	85
Table 47 TC010: Event Creation - Time Range Validity 1.....	86
Table 48 TC011: Event Creation - Time Range Validity 2.....	86
Table 49 TC012: Event Creation - Location Validity 0.....	87
Table 50 TC013: Event Creation - Location Validity 1.....	88
Table 51 TC014: Joining Event - Handling Payment 0.....	88
Table 52 TC015: Deleting Paid Event	89
Table 53 TC016: Deleting Free Event	90
Table 54 TC017: Event Streaming from Android phone - Account Validity 0.....	91
Table 55 TC018: Event Streaming from Android phone - Account Validity 1.....	91
Table 56 TC019: Event Streaming from Android phone - Account validity	92
Table 57 TC020: Event Streaming from Android phone - Retrieving Event list.....	93
Table 58 TC021: Event Streaming from Android phone - Retrieving Event list.....	93
Table 59 TC022: Event Streaming from Android phone - Streaming an Event.....	94
Table 60 TC023: Q&A section on the web - Start Instant Messaging	95
Table 61 TC024: Q&A section on the web Live Streaming over	95
Table 62 TC025: Q&A section on the Android - Start Instant Messaging.....	96
Table 63 TC026: Q&A section on the Android - Live Streaming over	97
Table 64 Estimated risk and contingency plan	98

List of Figures

Figure 1: Phases of Feature Driven Development	11
Figure 2: Ghannt Chart	12
Figure 3: Architecture of Android OS	16
Figure 4: RTP Packet Header.....	18
Figure 5: UseCase diagram of Thydash.....	24
Figure 6: Use Case diagram of User Management subsystem.....	25
Figure 7: Use Case diagram of Event Management subsystem	26
Figure 8: Use Case diagram of live streaming subsystem	26
Figure 9: Sequence Diagram for Create Account	28
Figure 10: Sequence Diagram for Create Event	29
Figure 11: Sequence Diagram for Join Event	32
Figure 12: Activity Diagram for Broadcast Live event	34
Figure 13: Sequence Diagram for watch live event	36
Figure 14: State Diagram for Download past events	37
Figure 15: Class Diagram	41
Figure 16: Object Diagram for Commenting on an event.....	42
Figure 17: Object Diagram for discussion	43
Figure 18: Object Diagram for creating an event.....	43
Figure 19: Object Diagram for joining an event	44
Figure 20: High Level Architecture of thydash	46

Figure 21: Django MTV Architecture.....	47
Figure 22: thydash Android application Architecture.....	48
Figure 23: Subsystem Decomposition of thydash.....	48
Figure 24: Hardware Software Mapping	49
Figure 25: Component Diagram	50
Figure 26: Deployment Diagram	51
Figure 27: UseCase Diagram for Boundary Conditions	52
Figure 28: Database Design	56
Figure 29: Web User Interface.....	58
Figure 30: Android User Interface	59
Figure 31MVC class diagram for User Management	63
Figure 32 Screenshot for Profile Page.....	64
Figure 33 MTV model for Event Management	65
Figure 34 Screenshot for Create Event	66
Figure 35 Screenshot for Event Details of Paid Event.....	66
Figure 36 Screenshot for Event Detail of Paid Event	67
Figure 37 MTV model for Streaming Module	68
Figure 38: Screenshot Live Events List	69
Figure 39 MTV Module for Instant Message module	71
Figure 40 Model of Document Relationship	73
Figure 41 Test case alignment with Test phases.....	80
Figure 42: Code Snippet - Event Model	101

Figure 43: Code Snippet - Create Event Template	102
Figure 44: Code Snippet - Insert Broadcast Method.....	102
Figure 45: Code Snippet - Chat Message class.....	103
Figure 46: Code Snippet - VideoStreaming connection	104

Acronyms

ICT: Information communication Technology.

UCD: Use Case Description

HTTP: Hypertext Transfer Protocol

RTSP: Real Time Streaming Protocol

RTP: Real Time Transfer Protocol

FDD: Feature Driven Development

P2P: Peer to Peer

IDE: Integrated Development Environment

IP: Internet Protocol

UDP: User Datagram Protocol

TCP: Transmission Control Protocol

RTMP: Real Time Messaging Protocol

Abstract

In the modern world, people in the cities live a pretty complicated life. Although there are so many inventions, innovations and technologies to simplify our life, we still are getting busier and busier.

We have a desire to relax, have fun and travel the world. That's why tourism is claimed to be a big industry with a big purpose. But the traditional way of touring is not very economical in a way that it takes time and money (transportation, food, shelter and others) of the tourist. What if we can tour somewhere just like going to cinemas? This is what we want to create.

To solve the desires of people to travel the world, even in weekends, we propose that live streaming based eTourism (live tourism) as an alternative solution, though it won't replace the traditional one.

[This page is intentionally left blank]

Chapter 1: Introduction

“You see only 936 full moons in your life time. You live through only 312 seasons and you get only 88 weeks of paid vacation”

1000 places to see before you die, Patricia Schultz

1.1 Background

In the last few decades, Information Communication Technologies (ICTs) have deeply affected the way business is done and the way that organizations compete. For instance, in tourism¹ industry, ICT has reshaped the entire process of tourism services development, management and marketing. It has improved the communication between tourists and tour operators, as well as the information tourists get about tour destinations. In addition to these, it plays significant role by facilitating creation of documentaries, websites, virtual reality technologies and many others.

Tourism can be either international or domestic to a country. International tourism has a huge effect on countries' income generation there by improving their GDP (Gross Domestic Product), accounting for one third of the total global services trade. These days; tourism is a major source of income for many countries in Africa and all over the world. Among these Kenya and South Africa are some of the main beneficiaries to mention. The effect of tourism isn't limited to economic sector only; rather it has direct effects on the social, cultural, and educational sectors of nations and international relations between them.

Though tourism has a hand full of benefits mentioned above, these days, it is becoming harder and harder since the world is facing lots of insecurities. A lot of tourists are being killed throughout the world. There are also terrorist activities that target tourists. In addition to this, many tourists who tour wild animals are being hurt and even dead. As a result of these insecurities to tourists, a tourist can't feel comfortable enough when going to a tourist location.

E-Tourism² has revolutionized consumer choice and service delivery processes in tourism industry. It maximizes their efficiency and effectiveness of tour operators, hotels, tourist attraction sites and other tourism related companies. Using eTourism travelers can obtain information about tourist destinations, choose the best ones that they want to visit, reserve hotels, reserve flight tickets etc. Thus, eTourism facilitates the process of visiting while increasing the satisfaction of travelers.

¹ “traveling to and staying in places outside their usual environment for not more than one consecutive year for leisure, business and other purposes” – according to WTO (World Tourism Organization)

² “analysis, design, implementation and application of IT and e-commerce solutions in the travel & tourism industry” – according to Wikipedia.com

This project aims to develop an eTourism platform, named **thydash**³, which helps people to explore the world through the internet using live video streaming from android based devices. Through thydash, eTour-guides will be able to create either paid or free events; eTourists⁴ can search for and join those events; on the time set for the event, eTour-guides⁵ will be able to live stream from their android devices where users who joined the event can attend it.

When planning to develop thydash, we claim that developing it will have a good impact in the tourism sector. In addition to that, it creates a good work place for unemployed people and it helps to improve their income.

1.2 Statement of the Problem

Tourism is a huge industry and the traditional way of touring is not very economical in a way that **it takes time and money** (transportation, food, shelter and many unmanaged minor things) of the tourist. Added to this, **disabled people** can't visit places easily. As a result, we thought there should be a way that could solve these problems, and here comes the idea of thydash.

From time to time the number of tourists around the world is increasing. People want to visit in their spare time, but our life style by itself is making us too busy now days. As a result, a way of tourism that puts all the tourists who can't travel to the actual location in to consideration will be much preferable this time. Hence we propose thydash as a solution to this real world problem.

Why thydash is better? It is better because it enables the tourist to participate in a tour wherever they are and whenever they want. As we have already stated, tourism is a huge industry and being able to participate in such industry could be beneficial. Automating tourism by itself comes with many advantages such as being able to tour regardless of location, time and mainly physically disabled tourists can tour easily.

As we can see, our world has many wonders. By building thydash, we can create an alternative way of touring the world. So, being able to implement thydash doesn't only solve the tourists' problems, it also enables the tour guide to have many global customers when compared to the traditional way of tourism.

³ **thydash** is a word coined from two English words; thy & dash. Thy meaning your and dash meaning moving quickly.

⁴ **eTourist** is a registered thydash user who attends events through thydash.

⁵ **eTour-Guide** is a registered thydash user who live streams events through thydash.

1.3 Purpose of the System

As we have described in the background part of this document, the tourism industry is losing a lot of tourists as a result of terrorism, insecurity, animal attacks and many others. As a result, if thydash is implemented it will create an alternative way to tour those places.

If thydash is implemented, it would be very useful in a way that it facilitates processes in the tourism industry, enables tourists who have little spare time to tour places using our platform, enables the tourists to save money, and enables physically disabled individuals to participate in tour events regardless of their location by using the internet.

1.4 Team Composition

The project team is composed of five students that are working together for the accomplishment of successfully implementing thydash. Generally, there is no task solely given to an individual. But here are the major responsibilities assigned for every group member.

Table 1: Team Composition

Name	Id	Email	Role
Yohannes Aregay	R/1898/04	emailtoyohannes@gmail.com	Chapter 1, Payment Implementation, Web User stories
Yidnekachew Wondimu	R/2388/04	searchyidne@gmail.com	Chapter 2, Chapter 4
Haben Birhane	R/0838/04	habenhadesh@gmail.com	Chapter 3, Chapter 4
Yeabsera Molla	03035771	bingoyab@yahooo.com	Mockup screens, Chapter 3
Melaku Minas	03035364	minasmelaku@gmail.com	Chapter 1, Web user stories, User Interface

1.5 Objectives of the Project

In this section we have described the general and specific objectives of developing thydash.

1.5.1 General Objectives

The main objective of this project is developing thydash that is alternative to traditional way of tourism. It solves the problems that the traditional way of tourism is facing like time, cost, physical incapability and others as described in the background section of this chapter.

1.5.2 Specific Objectives

- Allow individual tour guides to create events and stream their videos as per the schedule.
- Integrate thydash with available payment systems.
- Allow users to join and leave an event.
- Allow users to comment on tour events.
- Select the best way to implement live video streaming.
- Enable live video streaming from android based devices.
- Allow user to ask about the event using the question and answer session.
- Prepare an archive for live streamed videos.

1.6 Feasibility

In this section we explained feasibility of thydash using the four major yardsticks; technical feasibility, operational feasibility, economic feasibility and schedule feasibility.

1.6.1 Technical Feasibility

To implement thydash, the main technical requirement is having good internet connection. Thus, in areas where there is good internet connection thydash will perform better. Nowadays internet connection is becoming cheaper and faster. In addition to this android phones are also becoming smarter and cheaper.

Hardware without good software is of course nothing. To build thydash, there are lots of excellent open source projects which will either enable us learn how to implement it or become customized and used in developing our system, without the need to reinvent the wheel. Based on these reasons, we can decide that thydash will be technically feasible.

1.6.2 Operational Feasibility

Thydash will enhance the current tourism industry by creating many job opportunities; we say that for a reason. What anyone needs to be a tour guide in thydash is knowledge of how to use websites and mobile applications and should have android smartphone or camera. This makes it too easy to use it and generate income.

Since thydash will create an alternative way of tourism without replacing the traditional one, it won't result in workforce reduction, rather it will increase the income generated by the tourism industry. In addition, since we can schedule the time that we have, developing it won't conflict with other projects and activities. As a result, implementing thydash is operationally feasible.

1.6.3 Economic Feasibility

For thydash to be developed and deployed we need Django hosting web server, android devices and live streaming server. While we can use YouTube server for the live streaming, we are supposed to buy domain name and web hosting server and to use our android devices. As a result we can implement our system in much less cost.

1.6.4 Schedule Feasibility

There might be constraints like instability in the campus, illness of a team member and network problems. As a result, we might not perform as scheduled. But we will try to compensate the time wasted due to these reasons.

1.7 Scope and Limitation

In this section we have explained what will and what won't be included in our system.

1.7.1 Scope

In thydash, a user can either be an eTour-guide or an eTourist. The events created by eTour - guides can be either free or paid. Since we have limited resource and more importantly limited time, we chose to limit ourselves to the following areas:

- **User registration:** the users of the system will be individual tourists and tour guides.
- Registered users will be able to **create an event**.
- ETour-guides in thydash will be able to show tourist sites, cultural, social, religious and natural events by **live streaming videos** for users who joined the event.
- Live streamed events will be made available as an archive.
- ETourists will be able to discuss with eTour-guides while the event is undergoing.
- We will integrate thydash with already available payment system(s). Payments will be processed while joining a paid event.
- The platform will be accessed using web browsers and android devices.
- Users can download videos of an event from archives⁶ or memories⁷ either freely or by paying, depending on the interest of the event owner.
- The videos will be streamed from Android based devices i.e. smartphones and cameras

Optional features

- Allow users to follow the eTour-guides they like the most.

⁶ **Archives** are events that an eTour-guide has live streamed in the past.

⁷ **Memories** are past events attended by e-Tourists.

- Notify the user when there are new events from the eTour-guides they follow.
- Enable users to form a group and create group events.

1.7.2 Limitations

- A user can't upload a video for an event; rather thydash will only archive videos of live streamed events.
- We support live streaming **only from android based devices**.
- Internationalization is not supported in our system.
- We do not measure users' level of satisfaction for an event they have joined and attended.
- Our system is web based so we can't bring the feeling that the tourist feels by visiting physically.
- There should be good internet bandwidth to live stream videos smoothly.
- The android devices should have good camera quality to live stream quality videos.

1.8 Significance of the project

Thydash will play significant role in the tourism industry. In line with the traditional way of tourism, when our system is implemented fully, it will increase the number of tourists. Tourists who have attended events using thydash will be initiated to visit those places physically. There by increasing the amount of tourists and the income generated by the tourism industry.

Thydash will also have significant use for the tourists too, in a way that it brings them to the other side of the world wherever they are and whenever they want. And it will also reduce the time and cost they spend.

1.9 Methodology

To analyze and structure requirements of the system we will use tools that are used in agile software development methodologies.

One of the ways we will use is **Prototyping**⁸. We will iterate on our prototypes to gather the overall requirements of the system. For prototyping we will use a tool called Pencil.

In addition to this, we will prepare **User Stories**⁹ as a way to express the required functionalities of each features of the system. Then we will develop the system feature by feature iteratively.

⁸ A **prototype** is a draft version of a product that allows exploring ideas and showing the intention behind a feature or the overall design concept to users before investing time and money into development. A prototype can be anything from paper drawings (low-fidelity) to something that allows click-through of a few pieces of content to a fully functioning site (high-fidelity). – According to usability.gov

1.10 Development Tools

For the web application part, we are using the following tool sets:

- **Pencil:** mockups preparing (wire framing) tool
- **Python:** Programming Language
- **Django:** as a python web framework
- **MySQL client:** package for Django framework
- **Twitter Bootstrap:** as a front end web development framework
- **JQuery:** for a front end development
- **Windows, Linux (Ubuntu):** as a development environment machine
- **PyCharm:** as an IDE for the Django web framework
- **Microsoft Visual Studio, Enterprise Architect, Microsoft Visio :** as a diagramming tool
- **Microsoft Word:** for documentation
- **Microsoft PowerPoint, Prezi:** as presentation tool
- **Browser:** Mozilla, chrome
- **Git:** as a version controlling system
- **BitBucket:** as an online code repository

For the android application interface we are using the following tool set

- **Pencil:** as mockup preparing (wire framing) tool
- **Android studio:** as an IDE for android development

1.11 Testing Procedures

Developing software is a complex process. No matter how hard we try to eliminate all faults we cannot be so sure our system is error free. However, through good practice we can make sure that the most series fault does not occur in the first place. To simplify the testing process we followed the different types of tests that break the testing process up into the distinct levels. These types of testing are unit testing, integration testing, and system testing.

⁹ A **user story** is a tool used in agile software development to capture a description of a software feature from an end-user perspective. The user story describes the type of user, what they want and why. A user story helps to create a simplified description of a requirement. – According to searchsoftwarequality.techtarget.com/

1.11.1 Unit Testing

Unit testing¹⁰ is often automated but it can also be done manually. In developing thydash we will use automated unit testing for major functionalities and manual testing for the rest. In this section we will explain what functionalities we will test in each module of the system.

Unit test for user management module

In this module we will develop automated unit tests for login, registration, and logout.

- We will test login functionality of thydash by testing it whether it recognizes registered and unregistered user, validates a user for correct username and password.
- We will test logout functionality by checking if the module destroys the current session and redirects to login if any access to the system is required again.
- Added to these, we will test register (Create Account) functionality to check if it validates correct email input, rejects an already used username, validates the password for correct type and length and requires all the necessary fields in registration form.

Unit testing for Event management module

We will unit test this module for create event, manage event, join event and payment functionalities.

- We will test create event functionality of this module to check if it validates date and time is given to create an event, validates the correct format is given for price input and requires all the necessary fields for an event to be created.
- We will test manage event functionality to check if it requires privileges for accessing and enabled eTour-guides to update events that they created.
- In addition, we will test the join event functionality to check if it enables users, who have paid the amount of money required, to join an event.

Unit testing for Live streaming module

- In this module we will unit test effectiveness and efficiency of live streaming feature, check if the discussion session is created during event streaming, check that it doesn't allow any discussion session before or after event streaming, check a comment session is enabled after an event is live streamed.

¹⁰ **Unit testing** is a software development process in which the smallest testable parts of an application, called units, are individually and independently inspected for proper operation.

1.11.2 Integration Testing

Many times Integration testing¹¹ is done by inspection of the integrated components of the system. As a result, when we combine every units of the system, we will perform an integration testing by inspection.

Integration of User management module with Event Management module

When integrating user management and event management modules we will:

- Test if the user management module gives the required account information for event management module.
- Test if the user management module gives and requires permission for accessing event management module.
- Test if a user is registered before it watches or joins any event.

Integration of Event management module with live streaming module

When integrating event management and live streaming modules we will:

- Test if live streaming is done for an event that is already created.
- Test if the comment and discussion sessions are created for an event that already exists.
- Test if the event management gives all the necessary information for live stream module.

Integration of User management With Live stream module

When integrating user management and live streaming modules we will:

- Test if a user has joined an event before attending it.
- Test if a user is registered and logged in before he/she watches an event.
- Test if a user is registered and logged in before he streams an event.
- Test if the user management module has given all the necessary account information for live stream module.

1.11.3 System Testing

After unit and integration testing is undergone, there will be an overall system testing going on to check whether the overall system behaviors conform to the system requirements we proposed. Thus, the objective of system testing in thydash is to establish whether our application conforms to the requirements specified.

¹¹ **Integration testing** (sometimes called integration and testing) is the phase in software testing in which individual software modules are combined and tested as a group. It occurs after unit testing and before validation testing.

In system testing, we will check if the integration of all modules is complete and all the functional requirements are satisfied.

1.12 Overview of Project Phases

Since we follow Feature Driven Development¹² for our project the phases of the project will be like this:

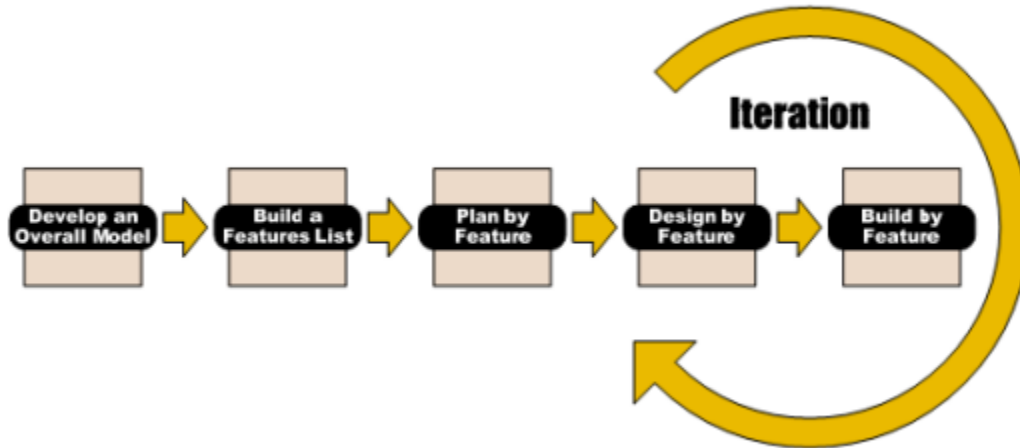


Figure 1: Phases of Feature Driven Development

Feature Driven Development has the following five phases

1. **Develop an overall model**- This phase contains development of the entire model of the project and overall object model.
2. **Build a features list**- This phase contains preparations of feature lists that will be implemented in the system.
3. **Plan by feature**- This phase contains planning on how to develop the features that are listed in the feature list.
4. **Design by feature**- This phase contains designing all listed features one by one.
5. **Build by feature**- This phase contains building and integrating all the developed features. The above two phases will be executed iteratively.

¹² **Feature-driven development** (FDD) is an iterative and incremental software development process. It is one of a number of lightweight or Agile methods for developing software.

1.13 Task and Schedule

Time schedule

Table 2: Time schedule

Task Name	Duration	Start	Finish
Prototyping	2 wks?	Mon 11/9/15	Fri 11/20/15
Documentation	3 wks?	Mon 11/23/15	Fri 12/11/15
UI Design	2 wks?	Mon 11/23/15	Fri 12/4/15
User Management	2 wks?	Mon 12/14/15	Fri 12/25/15
Create Event	1 wk?	Mon 12/28/15	Fri 1/1/16
Join Event	4 days?	Mon 1/4/16	Thu 1/7/16
Leave Event	2 days?	Thu 1/7/16	Fri 1/8/16
Live Streaming	8 wks?	Mon 1/25/16	Fri 3/18/16
Payment	4 wks?	Mon 1/25/16	Fri 2/19/16
Q&A	2 wks?	Mon 3/21/16	Fri 4/1/16
Search Events	2 wks?	Mon 4/4/16	Fri 4/15/16
Archive Events	1 wk?	Mon 4/18/16	Fri 4/22/16



Figure 2: Gantt Chart

Chapter 2: Literature Review

"Research is to see what everybody else has seen and to think what nobody else has thought"

Albert Szent-Gyorgyi¹³

Thydash is an innovative project which plans to create an alternative way of touring places by providing a platform where eTour guides can create a live tourism session to eTourists using live video streaming from an android powered devices i.e. camera and smartphones. While the eTourists tour the places of their interest, they can discuss with the eTour guide about anything they want to be clarified.

In this chapter, we have presented a literature review about how information technology is being used in the tourism sector and about technologies required to implement live video streaming from android based devices.

2.1 Information Technology in Tourism Sector

Due to the contribution of information technology, tourism nowadays is not the same as it was earlier. The times where tourists blindly choose places to visit and went uninformed about the overall conditions of the area has gone forever, and this is due to the availability of internet marketing, the virtual tourism¹⁴, documentaries and online maps such as Google map.

The ICT driven re-engineering has gradually generated a new paradigm shift¹⁵, altering the industry structure and developing a whole range of opportunities and threats (Bethapudi, 2013). Inevitably, Tourism is one of the industries that will mainly be affected by the paradigm shift.

The internet is an important means of promoting and commercializing tourist sites due to its globally reachable nature and multimedia capability. (Lois , Bill, & Joan, 2002). The multimedia capability of the internet has enabled tour operators to promote sites and it has made sharing tourist experiences easier. These shared images and media files act as a mediator where the viewers feel initiated to visit their area of interest. In addition to this, the internet changed the

¹³ **Albert Szent-Györgyi** was a Hungarian American physiologist who won the Nobel Prize in Physiology or Medicine in 1937. He is credited with discovering vitamin C and the components and reactions of the citric acid cycle

¹⁴ A **virtual tourism** is a simulation of an existing location, usually composed of a sequence of videos or still images. It may also use other multimedia elements such as sound effects, music, narration, and text.

¹⁵ A **Paradigm Shift** is a fundamental change in an individual's or a society's view of how things work in the world.

way of marketing (promotion) from general to an individualized one, and now it is made to be suitable to the interests of individual consumers. (Hawkins, Leventhal, & Oden, 1996)

Through e-Tourism, tourist organizations have changed the way they communicate with individual and organizational clients. Clients can undergo all the preceding processes online, i.e. they can reserve and pay for the services they get while touring. (Buhalis & Deimezi) .

2.2 Live Video Streaming from Android Devices

2.2.1 Overview

Streaming is a technique of transferring data which is processed as a steady and continuous stream whereby the client does not have to download the entire file to view it. Streaming Audio and video enables transferring data over long distances. The content to be streamed could be either accessed on demand or live.

Video streaming can be one-way or two-way. While one-way video streaming technology addresses larger audience it lacks interactivity whereas a two-way video streaming offers high level of interactivity at the expense of limited number of audiences. (Greenberg & Zanetis, 2012)

In video streaming, the end user can start viewing almost as soon as the video or multimedia data starts playing on the server. To achieve this, bandwidth efficiency between video servers and equipment of end users is very important. (Huifang Sun, 2007)

2.2.2 Live Video Streaming

In these days, the popularization of devices and continuous efforts to increase network quality has brought a proper environment for the rise of live streaming. We can create live streaming events either by using camera which is connected to a personal computer or by using smartphones. Due to the technological advancements made in the area of smart phones, nowadays we can get smartphones with quality cameras. This enables it easier to live stream moments from these handy smartphones.

2.2.2.1 Types of Live Video Streaming

There are two types of architectures which can be used for implementing live video streaming, i.e. **client-server** and **peer to peer**. We will discuss about both types in this section.

I. Client – Server

The two remote, internet connected platforms are the Android¹⁶ devices in this context and the server where the video is received. The android device logs on the server using a custom protocol over a TCP connection established on an Android device port of the server.

In this system, the android device turns the camera on and begins transferring the video over UDP. The video is sent as RTP packets to the server. Users with an RTSP capable video player or RTSP client applications wishing to view or decode the stream connect to the RTSP server. After the stream information such as encoding type and connection information is requested from the server, the client sends a play command.

When the server receives this request it will begin forwarding the RTP packets to the viewer. The user's RTSP client decodes the video and encodes it in an embeddable web video project. The encoding is saved to a Flash Video file on the server's hard drive or in memory.

II. Peer to Peer (P2P)

This is a type of Live streaming between mobile phones. i.e. It is done without the use of video processing servers or network infrastructure. (Hu, Nick, Jadwiga, & Marius) Currently a majority of smart phones are equipped with both hardware that supports live video processing and ad-hoc wireless communication between peers. This allows live video streaming between peer devices. (Hu, Nick, Jadwiga, & Marius)

The communication between mobile phones can be either between peer devices or over multiple hops between those devices. In creating multiple hops, phones within a communication range of each other automatically establish a wireless link creating a client mesh network. Each phone in the client mesh network is able to produce, consume and forward video to other neighboring devices.

To stream a multimedia file across multiple hops, a multi-hop path between the sender and the receiver has to be found. A routing protocol is therefore required to establish and manage connectivity within the mobile ad hoc network. A routing protocol may form various logical network topologies depending on the selected metrics (such as link quality) and Quality of Service (QoS)¹⁷ requirements.

¹⁶ **Android** is a mobile operating system (OS) currently developed by Google, based on the Linux kernel and designed primarily for touch screen mobile **devices** such as smartphones and tablets

¹⁷ **Quality of service (QoS)** is the overall performance of a telephony or computer network, particularly the performance seen by the users of the network. To quantitatively measure quality of service, several related aspects of the network service are often considered, such as error rates, bit rate, throughput, transmission delay, availability, jitter, etc.

Another requirement in P2P live streaming is discovery of video content providers in the ad-hoc networks. There are two ways to discover video content providers, one is periodic broadcast advertisement and the other is publish subscribe model.

Video frames need to be encoded before sending onto network. Encoding can significantly reduce the size of individual video frames, therefore minimizing the bandwidth required for the streaming application. The encoded video frames will be decoded in the receiver side before being displayed on the screen.

2.2.2.2 Android OS

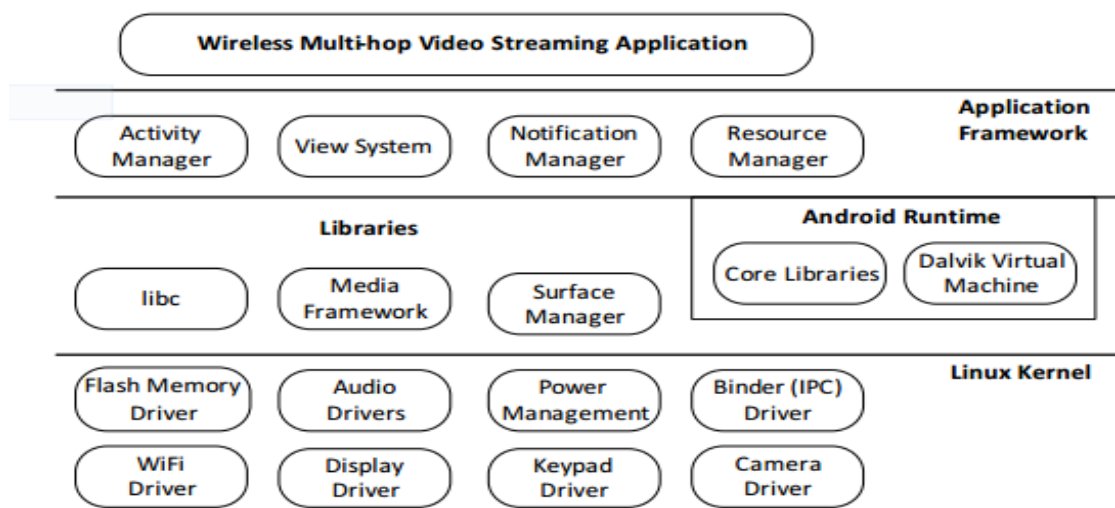


Figure 3: Architecture of Android OS

source (Bailey, 2011)

Android applications are mostly written in Java and run on Dalvik, Android's own Java virtual machine. As it can be seen in the Android System Architecture, it is built based on the Linux kernel with device drivers and the Android runtime environment (along with a number of libraries) that support interactions between the Linux kernel and the high-level application framework.

In addition to the SDK, there is also a native development toolkit (NDK), which supports the use of native C or C++ codes in the applications. The NDK is an extension of SDK to allow the development of lower-level source codes for more efficient data processing in the system.

2.2.2.3 (En|De) Coding

Video encoding and decoding is an important aspect of any video streaming application. The encoders and decoders have the role of transforming the live video stream from the RTSP encoded video into Flash video using one of the available CoDec formats. There are many ways to encode and decode a video, and each type has its own advantage and disadvantage when compared to other. The differences between consist of various properties such as the file size, encoding time, decoding time, processor usage, memory footprint, video quality and many more. (Bailey, 2011)

In live video streaming application, the performance of video encoding and decoding is an indication of the feasibility of the application. The speed of the en/decoding has an impact on perceived quality of the video which intern affects the usability of the application. This means, to transfer a video from an Android device, an efficient encoder and decoder is desired for optimal scalability.

Basically, encoding can be classified into inter frame encoding and intra frame encoding.

- **Interframe** encoding is the simplest form of encoding. It treats every frame as an individual image to encode. This method is resilient against lost frames due to each frame having enough information to create an entire image. (Hu, Nick, Jadwiga, & Marius)
- **Intraframe** encoding uses two types of frames, i.e., the key frames and predicted frames for better compression ratio. They key frame contains complete information to create an image, whereas the predicted frames only contain the differences between key frames thus previous frames area required for their successful decoding.

2.2.2.4 Codec Formats

A Compressor De Compressor, or CoDec, format is used for compressing and decompressing digital video, thus converting raw (uncompressed) digital video to a compressed format or vice-versa. A variety of video CoDec formats can be implemented in encoding and decoding a video. Here is a list of widely used codec formats:

Motion JPEG (MJPEG)

MJPEG is a form of intra frame encoding that encodes each frame as a JPEG image.

MPEG – 4

MPEG – 4 supports both inter frame and intra frame encoding and also allowing profiling for devices with different resource capabilities.

H.263

H.263 is a video encoding and decoding standard that was originally developed in 1996 by ITU-T video coding experts group for low bandwidth videoconferencing with adjustments made for higher quality streams in 1998 and 2000. (Bailey, 2011)

H.264

H.264 is a popular coding technique that has is being used by a number of online multimedia services, such as YouTube. Similar to the MPEG-4 encoding, H.264 allows profiling according to the needs and capabilities of various devices. (Hu, Nick, Jadwiga, & Marius)

2.2.2.5 Internet Protocols

TCP is the dominant protocol for data transfer over the internet. Of course, TCP can be used for video streaming, but several problems needs to be addressed in order to provide a quality video streaming. (Huifang Sun, 2007) The first is how to handle the data rate variability and the delay in end-to-end retransmission of videos will be the second. This can be solved by using buffers. The third problem is how it handles packet loss. If packets are lost during the transmission, it stops working until the lost packet arrives or is retransmitted. In addition, it doesn't allow a multicast. And those problems mentioned above make TCP unsuitable for video streaming.

UDP is another protocol that can be used in video streaming. This protocol holds packet loss in a different way than the TCP. It allows packet loss where the user notices damages in the video transmission, but the transmission continues rather than seeking a retransmission.

2.2.2.6 RTP

RTP packet header							
Bit offset ^[b]	0–1	2	3	4–7	8	9–15	16–31
0	Version	P	X	CC	M	PT	Sequence number
32	Timestamp						
64	SSRC identifier						
96	CSRC identifiers						
	...						
96+32×CC	Profile-specific extension header ID					Extension header length	
128+32×CC	Extension header						
	...						

(Source: Wikipedia)

Figure 4: RTP Packet Header

Real-time Transport Protocol, or RTP, is a packet format for transmitting audio and video encoded with various encoder types over the internet or local networks. (Bailey, 2011) It was developed by the Audio-Video Transport Working Group of the Internet Engineering Task Force (IETF).

RTP is designed for end-to-end, real-time transfer of streamed multimedia file. The protocol supports jitter compensation and detection of out of sequence arrival in data, which are common during transmissions over an IP network. It is transmitted over UDP, thus it does not require TCP acknowledgements, or TCP ACK packets.

Many software and hardware applications support RTP, such as Windows Media Player, QuickTime, VLC, mplayer, and ffmpeg. The applications may call upon other decoder software installed in the system, to play the video contained within RTP.

2.2.2.7 RTSP

The Real Time Streaming Protocol, or RTSP, is used to control the playback of an RTP stream, i.e. An RTSP Server allows a viewer to receive the RTP packets sent from the device. A viewer (e.g. VLC or ffmpeg) cannot directly open an RTP stream, therefore an RTSP server is required to provide an additional stream information and control to start and stop the streaming packets. The RTSP server is based on RFC 2326, published in 1998, and It incorporates the following commands: options, describe, announce, setup, play, pause and teardown. (Bailey, 2011)

RTSP breaks the large data into packet sized ones based on the bandwidth available between the source and destination. This allows for the client software to play one packet, while decompressing the second packet and downloading the third. Users would listen or see the media files without feeling a break between the data files.

2.2.2.8 Flash Player

Since the FLV file is outputted by the encoder, the entire video recording will be saved to this file. If the video were to be played directly through the HTTP server, this will result in the video played being the beginning of the recording not actually being the live video. To combat this problem, after the live streamed video is decoded by the receiver it can be played using flash video player.

2.2.3 Live Streaming Providers

There are many live streaming service providers. Among those, we have selected three service providers based on their popularity nowadays. These are the live version of YouTube, Twitch, and Periscope.

2.2.3.1 YouTube Live

YouTube Live is the live streaming service owned by Google Inc. Offered as the live branch of the biggest Video on Demand (VoD) service, YouTube. It provides live support to global events, such as popular concerts, sports events, and interviews. (Pires, 2015)

The videos in YouTube are encoded according to the H.264/MPEG-4 Advanced Video Coding (AVC) as default videos compression format. (Wamser, Hock, Seufert, Staehle, Pries, & Tran-Gia, 2013) Thus live streaming clients should use these CoDecs to live stream in this platform.

YouTube uses progressive HTTP streaming technology. The client essentially downloads the video data over an HTTP connection, and already starts playing the video while the download is yet to be completed. The client application is a precompiled Adobe Flash player assembly that runs at the client in the web browser. Downloaded data is stored in a temporary file, for the server to buffer in video playtime.

2.2.3.2 Twitch

Twitch is a live video streaming platform owned by Amazon Inc. It first appeared in June 2011, and it's primarily focuses are on video gaming and broadcasts of eSports competitions. Contents of this site can either be viewed live or on demand based on the broadcaster choice of settings. In the live streaming session, viewers can interact with the broadcaster and other viewers by the chat aside the video.

2.2.3.3 Periscope

Periscope is a live video streaming app for IOS and Android, developed by Kayvon Beykpour and Joe Bernstein. They started the company in February 2014, under the name Bounty. In January 2015, it was owned by Twitter Inc.

Periscope allows viewers to send hearts to the broadcaster by tapping on the mobile as a form of appreciation. Scopes disappear from the website after 24 hours, but users can capture their scopes using other applications.

Among the above service providers, YouTube provides a live streaming API which external applications can use for their purpose. Thus, in the case of ours, we have selected to integrate YouTube Live streaming service with our android application which will encompass

Chapter 3: Proposed System

“Creativity is just connecting things. When you ask creative people how they did something, they feel a little guilty because they didn’t really do it, they just saw something. It seemed obvious to them after a while”
-- Steve Jobs¹⁸

3.1 Overview

Thydash is an eTourism platform based on a live video streaming and discussion. It is proposed as an alternative for the traditional tourism to help impaired and financially inefficient part of the society on one hand and fascinated travelers on the other. This product will be a **new** software product for **e-Tourists** and **e-Tour Guides** all around the world.

The main purpose of this chapter is to discuss about the proposed software product, in terms of its functional requirements, non-functional requirements, the user and system interaction using Use Case diagrams, the Object relations using dynamic models and the general object model of the product.

3.2 Functional Requirements

In this section of the documentation, we will discuss about the main functionalities/features of thydash. We will describe each functional requirement of the product under its subsystem decomposition for ease of understanding.

User Management - this subsystem handles the information about users of thydash, every user will have a unique identification number. This subsystem includes:

- I. **Account Creation** - this is the functionality of the product which handles the task of creating a new user.
- II. **Log in/ Log out** - this functionality of the system is used to authenticate the users of thydash.
- III. **Profile** - users will be able to fill and update their account information.

Event Management - this subsystem handles, issues related to Events. Every event will have a unique identification number. This subsystem includes:

¹⁸ **Steve Jobs** was an American information technology entrepreneur and inventor. He was the co-founder, chairman, and chief executive officer of Apple Inc.

I. Create Event - this is the main functionality of the Event management subsystem, and will be used by e-Tour Guides to create an event by filling the required information, including the event's promo video.

II. Delete Event - is used by the e-Tour Guide to delete an already created event, and this will send notification to the e-Tourists about the update and invokes the payment handler if it was paid event.

III. Join Event - this functionality is used by e- Tourists to join to an event, and this may invoke the payment handler if it is a paid event

IV. Leave Event - this is the functionality e-Tourists use if they want to quit the event that they have joined. This may invoke the payment handler accordingly.

V. Archives/ Memories - is an instance of archive where eTour-guides/ eTourists will find their past events in an organized format.

VI. Comment - this feature of Thydash is designed for e-Tourists to write a comment, an event after it is completed.

VII. Download from Memories – users will be able to download videos of events from their memories, events that they have participated in the past.

Live Streaming - this is the subsystem where we are going to handle issues related to live video streaming, and this it includes:

I. Broadcast Live - is one of the main functionalities of the thydash, it is used to broadcast live events using live video streaming to other e-Tourists.

II. Watch Live - is one of the main functionality of the subsystem and the product as a whole. E-Tourists will use this functionality to attend a live event.

III. Q & A (Discussion) - this is among the important features of our system. Users in a live event can discuss their ambiguity with e-tour guide using this facility.

IV. Auto Archive - this feature of thydash is designed to automatically save videos of live events after transmission is finished.

Payment- this functionality of our system handles every money transactions made by any user while joining, leaving, deleting and finishing an event.

3.3 Non Functional Requirements

In this section, we will try to specify criteria that can be used to judge the operation of our system, other than functional behaviors. We have classified the non-functional requirements in to quality requirements and constraints. The quality requirements of thydash are the following:

Usability Requirements

- Users of the system should be able to walk through the features of thydash without a professional aid from others.
- The user interface should be simple, easy to use, and easy to learn for an average web and android user. The system should interact with the user in a simple and efficient way.

Security Requirements

- User information is the most crucial element of our system. Thus, we will implement different security measures, such as encryption techniques, to protect users' important information from malicious attack.

Reliability Requirements

- Thydash, when implemented, should show erroneous inputs and the current state of the system. In addition, it should be able to recover data and revive to its initial state if deadly situations happen to the system.

Performance Requirements

- Availability - the system should be available every time, except for maintenance purposes.
- Response time - our system is expected to be pleasant by responding to the user requests in reasonable time on areas where there is a good internet connection.

Supportability Requirements

- Changes made to some part of thydash after it is deployed shall not have an impact to the whole product; the system should be designed to be loosely coupled.
- It should be easy to reuse or change components of the system.

Here we have listed constraints or pseudo-requirements of thydash:

Interface Requirements

- Thydash complies with the protocols and other software interfaces specified by YouTube Live API.
- Thydash complies with the API endpoints defined by payment processing systems such as Braintree or PayPal.

- Thydash complies with the interfaces provided by Google Maps API and Google Auth API.

Legal Requirements

- Though there are multiple legal requirements to implement this system in real world, we won't handle these issues at the end of the educational year. Those issues will be handled in the next versions of the system.

3.4 System Model

In this section, we will discuss and describe the conceptual model that describes thydash. The different views of the system will be described using different UseCase diagrams and use case descriptions.

3.4.1 UseCase Model

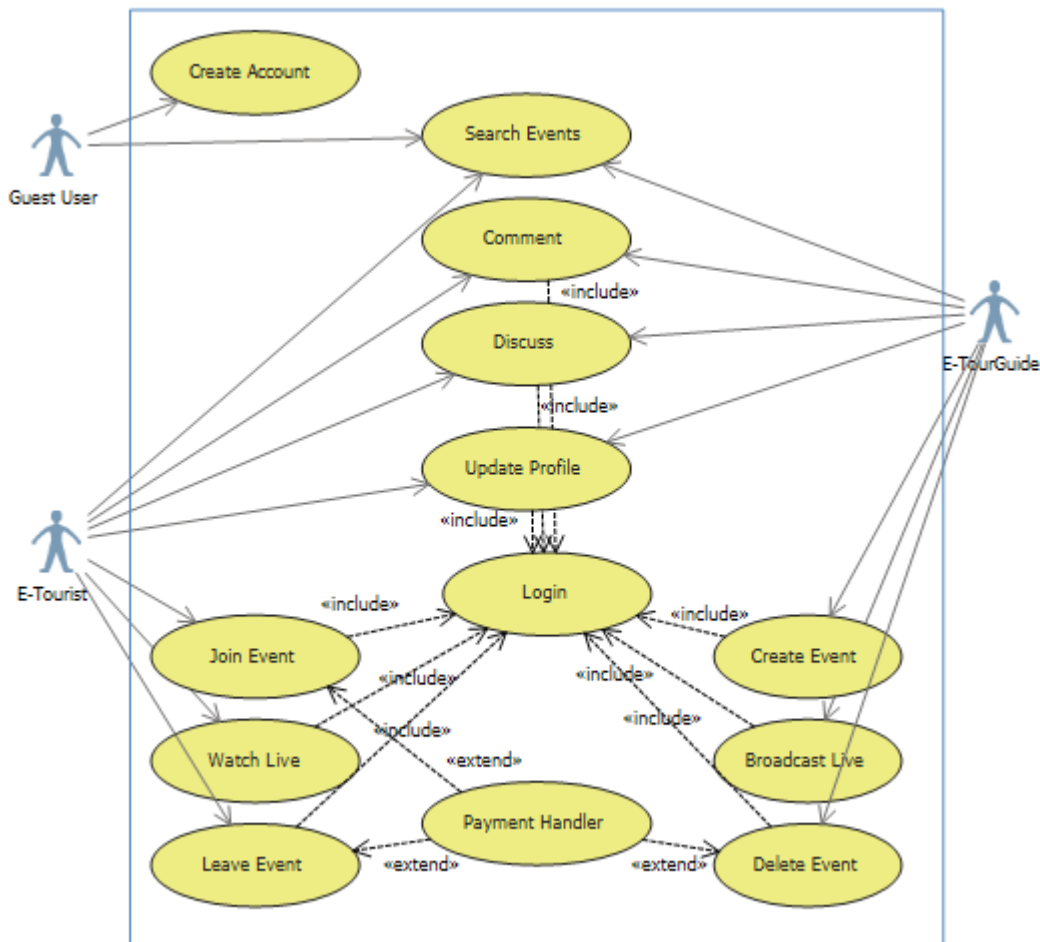


Figure 5: UseCase diagram of Thydash

Use Case model of Subsystems of Thydash

The General Use Case model of Thydash is described in subsystem format for the ease of readability and description. Therefore, the subsystems identified for this section are user management, event management and live video streaming. They are presented as follows.

User Management Subsystem - mainly describes the relationship between thydash users and the functionality under this subsystem.

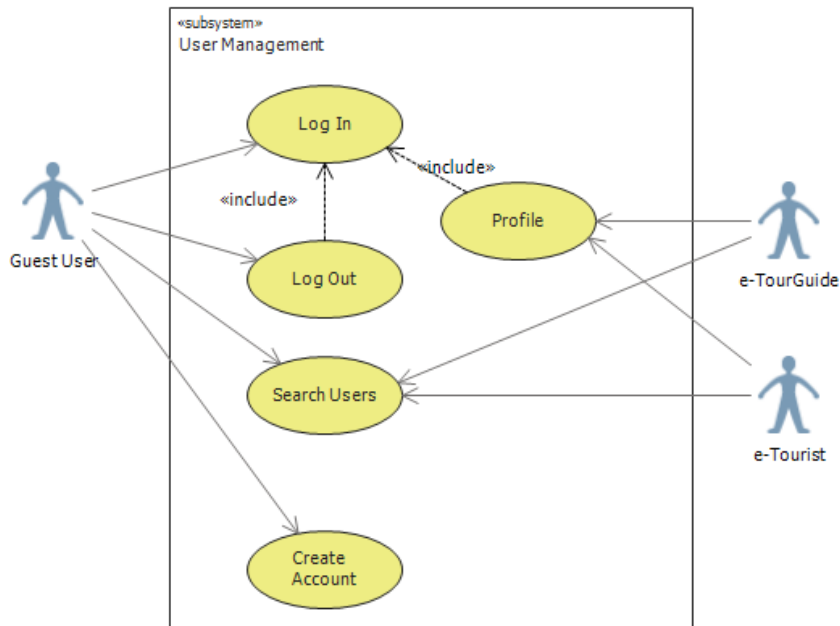


Figure 6: Use Case diagram of User Management subsystem

Event Management Subsystem - mainly describes the relationship between Thydash users and the functionality under this subsystem, i.e. Create Event, Delete Event, Join Event and Leave Event. All these functionalities include the log in use case, because to perform all the tasks mentioned above, the e-Tourist/e-Tour Guide needs to login first.

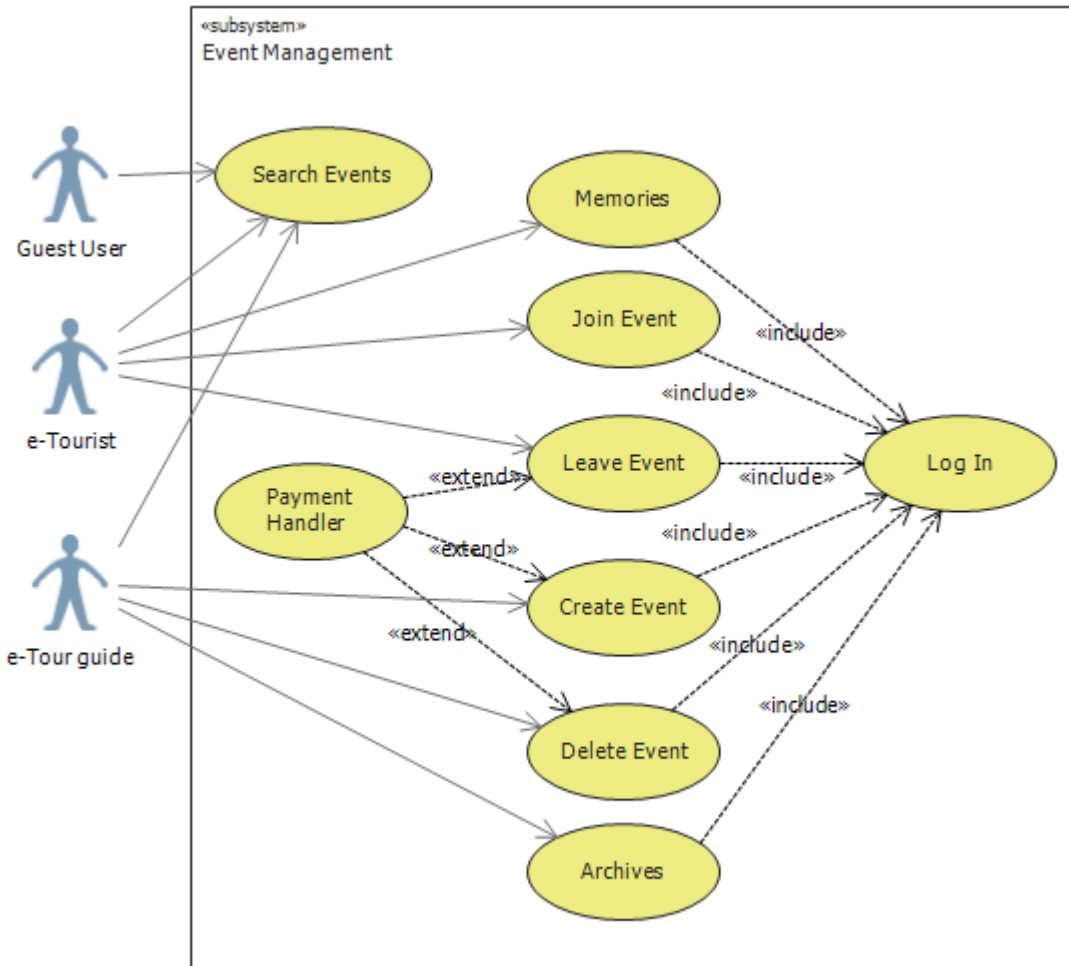


Figure 7: Use Case diagram of Event Management subsystem

Live streaming subsystem - All the tasks related to the live video streaming are handled in this subsystem.

uc UMLUseCaseDiagramStreaming

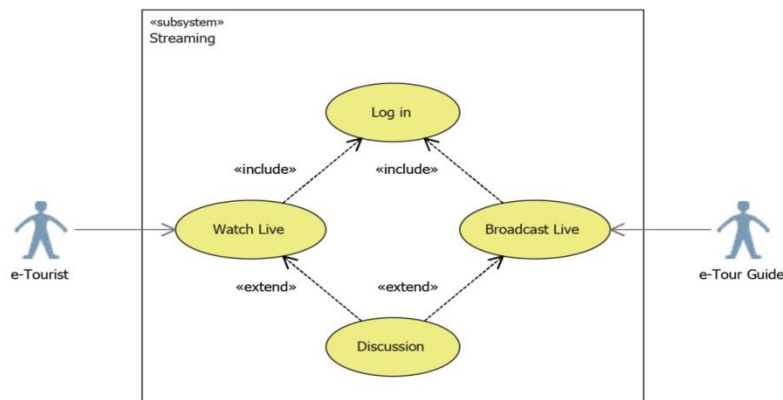


Figure 8: Use Case diagram of live streaming subsystem

3.4.2 UseCase Description

For the textual description of our use cases, we use a template composed of six fields(Use Case Name, Use Case No., Description, Participating Actors, Entry Condition, Flow of Events, Exit Condition) adapted from [Constantine & Lockwood, 2001]

In addition, we have included dynamic models that describe the use cases.

Table 3: UCD for Create Account

Use Case Name	Create an Account
Use Case No	UC01
Description	Registering a new user in the Thydash platform
Participating Actors	Guest User
Entry Condition	1. There is a decent internet connection
Flow of Events	<ol style="list-style-type: none"> 1. User writes the following web address in the URL name service. <u>www.Thydash.com</u>. 2. The guest user page will show up and the guest user clicks in the Create Account link. 3. The user registration form will be displayed, and guest user will fill the form with valid information and clicks on the Create Account button and automatically the system sends confirmation code to the users email. 4. The guest user needs to activate his/ her Thydash account by clicking in the validation key link in his/her email address. 5. The System will display a success confirmation message.
Exit Condition	1. User will have Thydash account.
Alternatives	<ol style="list-style-type: none"> 1. Guest User fills didn't activate his/her account. <ul style="list-style-type: none"> • System requests the user to activate its account. 2. Guest User clicks on an expired activation key(after 7 days) <ul style="list-style-type: none"> • System rejects the activation key and sends new activation key to Guest User's email.

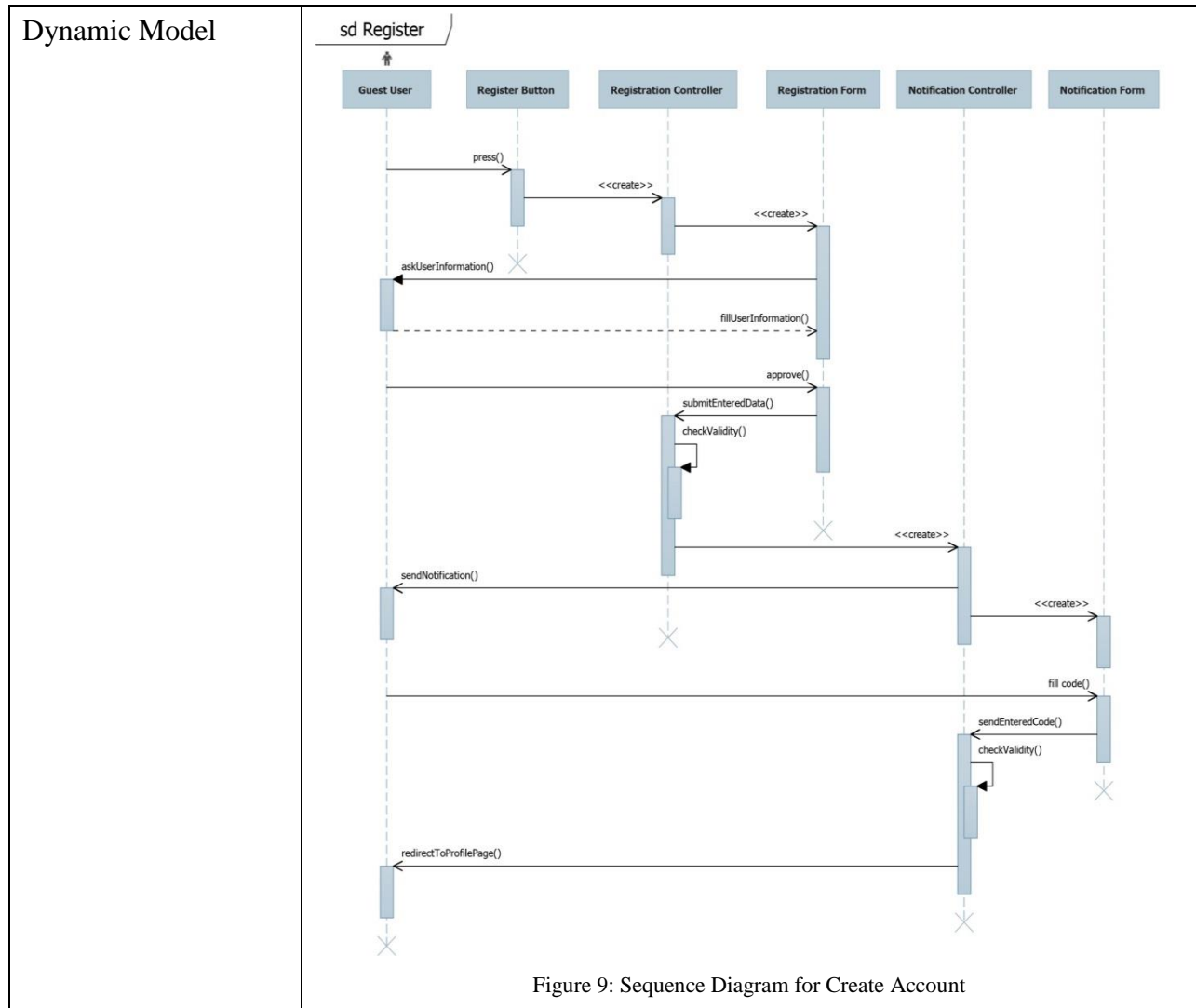


Table 4: UCD for Create Event

Use Case Name	Create Event
Use Case No	UC02
Description	eTour-guide logged in, so that he/she can make an event to their subscribers.
Participating Actors	e- Tour Guide
Entry Condition	1. e-Tour guide logged in to his/her Thydash account.
Flow of Events	2. User clicks on the Create Event link. 3. User will be forwarded to the page where the form to create an event is

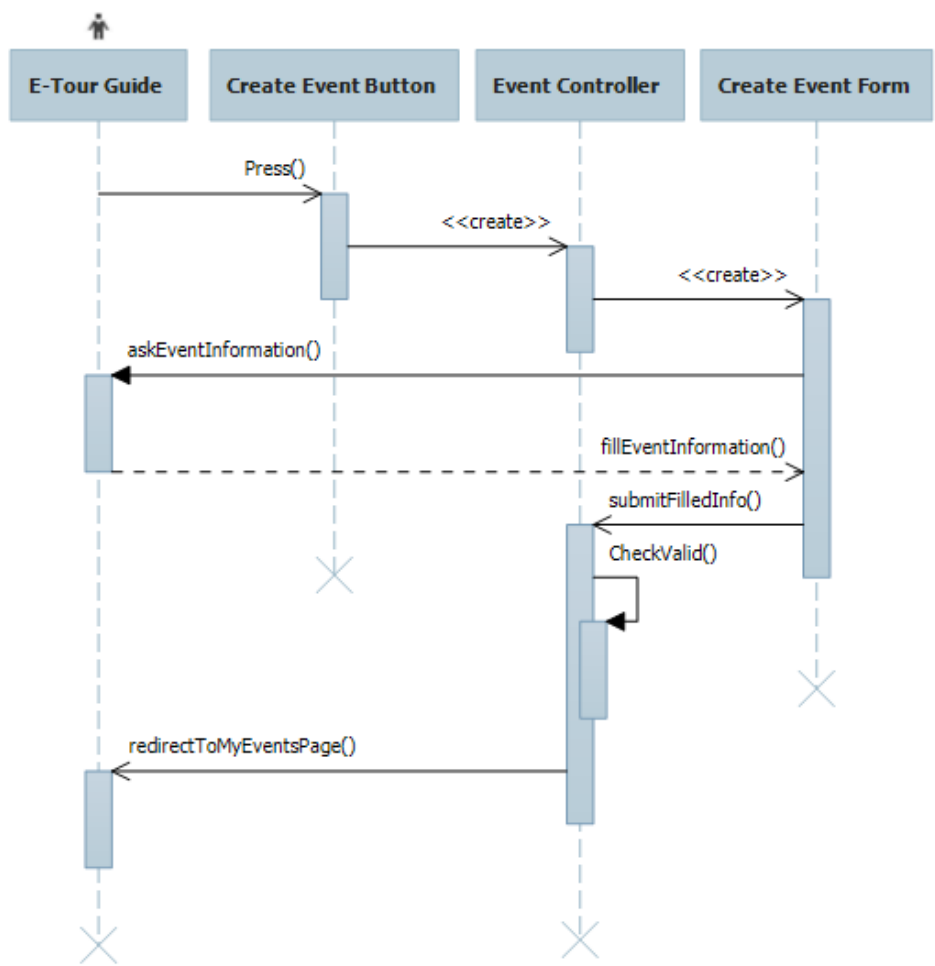
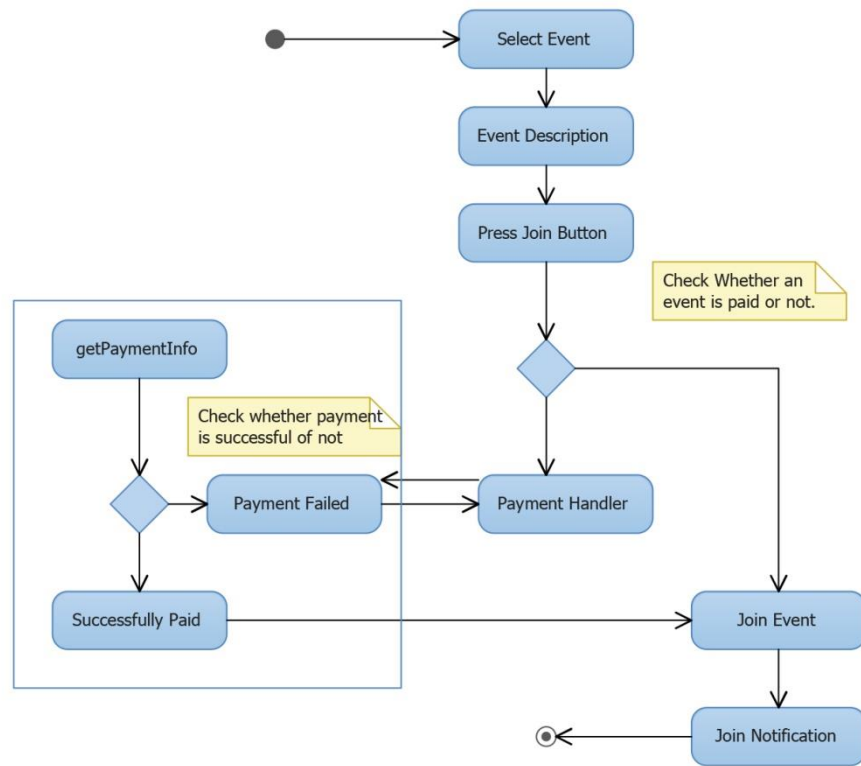
	<p>found.</p> <p>4. User fills the necessary event information such as, event name, promo image and video, price, time and duration then clicks on the Save Event button.</p> <p>5. The system will send a success confirmation message and forwards the user to the My Event page.</p>
Exit Condition	1. e-Tour Guide has created an event.
Alternatives	<p>1. User didn't fill all necessary information.</p> <ul style="list-style-type: none"> System sends an error message and asks user to refill the form with correct Event information.
Dynamic Model	 <pre> sequenceDiagram actor User participant E-Tour Guide participant Create Event Button participant Event Controller participant Create Event Form User->>Create Event Button: Press() activate Create Event Button Create Event Button->>Event Controller: <<create>> deactivate Create Event Button activate Event Controller Event Controller->>Create Event Form: <<create>> deactivate Event Controller activate Create Event Form Create Event Form->>E-Tour Guide: askEventInformation() deactivate Create Event Form activate E-Tour Guide E-Tour Guide-->>Create Event Form: fillEventInformation() deactivate E-Tour Guide activate Create Event Form Create Event Form->>Event Controller: submitFilledInfo() deactivate Create Event Form activate Event Controller Event Controller->>Event Controller: CheckValid() deactivate Event Controller Event Controller->>E-Tour Guide: redirectToMyEventsPage() deactivate Event Controller deactivate E-Tour Guide </pre> <p>The sequence diagram illustrates the process of creating an event. It involves four main components: E-Tour Guide, Create Event Button, Event Controller, and Create Event Form. The process begins with the user pressing the 'Create Event Button', which triggers the 'Press()' message. This leads to the creation of the 'Event Controller' and then the 'Create Event Form'. The 'Create Event Form' then asks the 'E-Tour Guide' for event information ('askEventInformation()'). The 'E-Tour Guide' fills in this information ('fillEventInformation()') and submits it to the 'Event Controller' ('submitFilledInfo()'). The 'Event Controller' performs a self-check ('CheckValid()') and then redirects the user to the 'My Events Page' ('redirectToMyEventsPage()'). The diagram also shows a dashed line representing an alternative path where the user might not fill all necessary information, leading to an error message and a request to refill the form.</p> <p>Figure 10: Sequence Diagram for Create Event</p>

Table 5: UCD for Join Event

Use Case Name	Join Event
Use Case No	UC03
Description	An e-Tourist logged in, so that he/she can join an event.
Participating Actors	e-Tourist
Entry Condition	1. e-Tourist logged in to his/her Thydash account.
Flow of Events	<ol style="list-style-type: none"> 1. User selects event using the upcoming event page or from the home page. 2. User will be forwarded to the page was he/she can check the detail of the specific event. 3. By checking the detail, if the user is interested, then he/she clicks in join event button. 4. The system sends a success confirmation message.
Exit Condition	1. e-Tourist has successfully joined an event.
Alternatives	<ol style="list-style-type: none"> 1. e-Tourist is not interested after checking the detail of the <ul style="list-style-type: none"> • E-Tourist aborts the join event option and returns it Home page. 2. The event is a paid event; the payment handler will be invoked automatically. <ul style="list-style-type: none"> • e-Tourist selects an payment handler and submits a valid account number • The system sends a success confirmation message. 3. User didn't gave a valid account information. <ul style="list-style-type: none"> • The system will abort the transaction and sends a message to e-Tourist to enter a valid account information.

Dynamic Model

act join_event



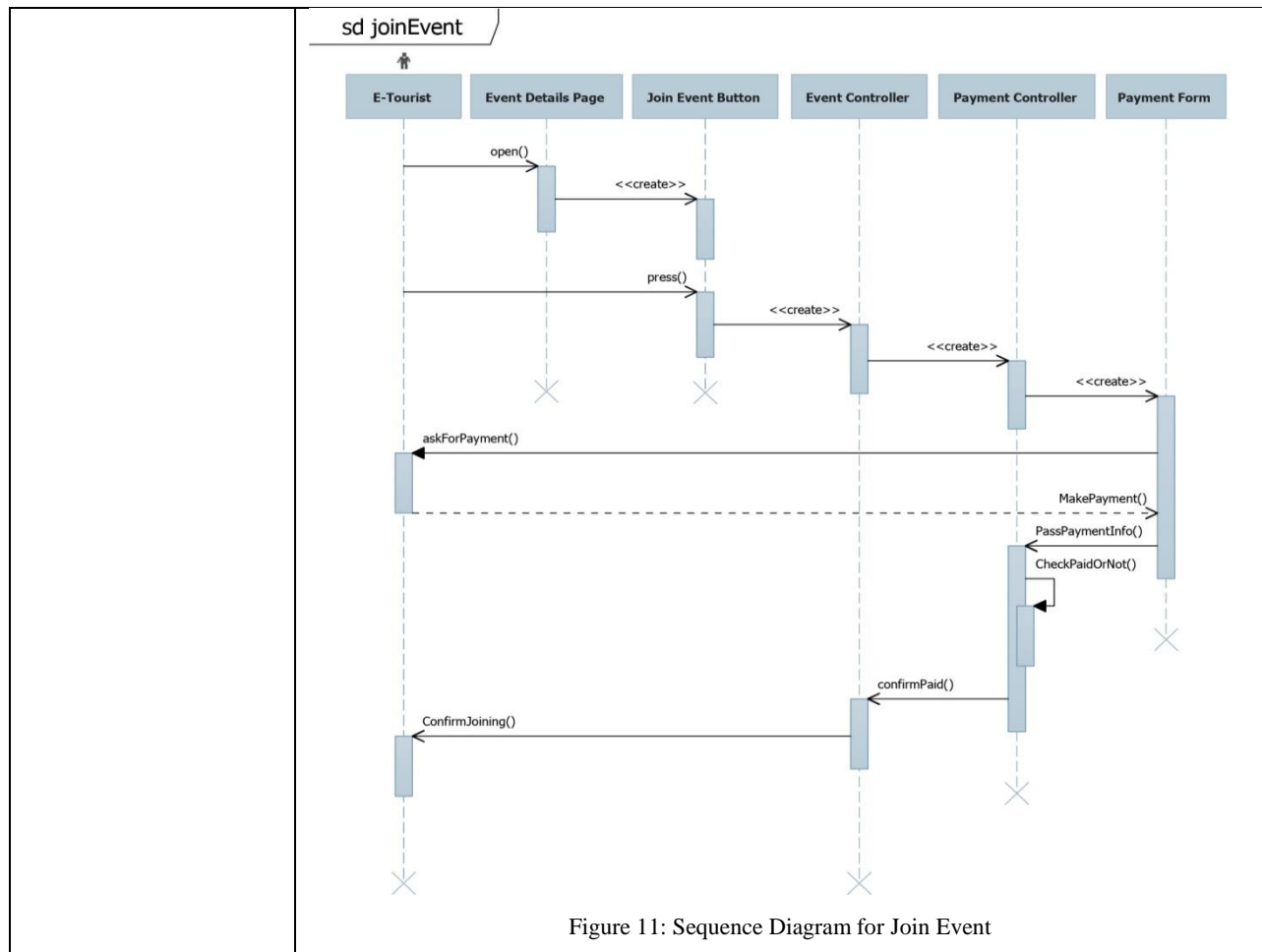


Table 6: UCD for broadcast live event

Use Case Name	Broadcast Live Event
Use Case No	UC04
Description	An e-Tour Guide wanted to make broad cast an event live to e-Tourists who Joined the event he/she created.
Participating Actors	e- Tour guide
Entry Condition	1. e- Tour guide logged in to his/her Thydash account using its android device. 2. e- Tour guide created an event.
Flow of Events	1. e- Tour guide selects the event that he/she is prepared to broadcast from the list of events he made. 2. e- Tour guide can check the status of the event such as, number of e-

	<p>Tourists who joined the event, the time remaining for the proposed broadcast.</p> <p>3. e- Tour guide clicks on the Make Live button when he/she decides to start broadcasting.</p> <p>4. The system starts connecting to the YouTube live streaming server, and e- Tour guide starts broadcasting the live event.</p> <p>5. e-Tour guide answer the questions in the Q&A (Discussion).</p>
Exit Condition	1. e- Tour guide has successfully finished live streaming the event.
Alternatives	<p>1. The system faces a connection error, or couldn't make a connection with the YouTube live streaming server.</p> <ul style="list-style-type: none"> • system retries to make that connection again.

Dynamic Model

act LiveBroadcasting.thydash

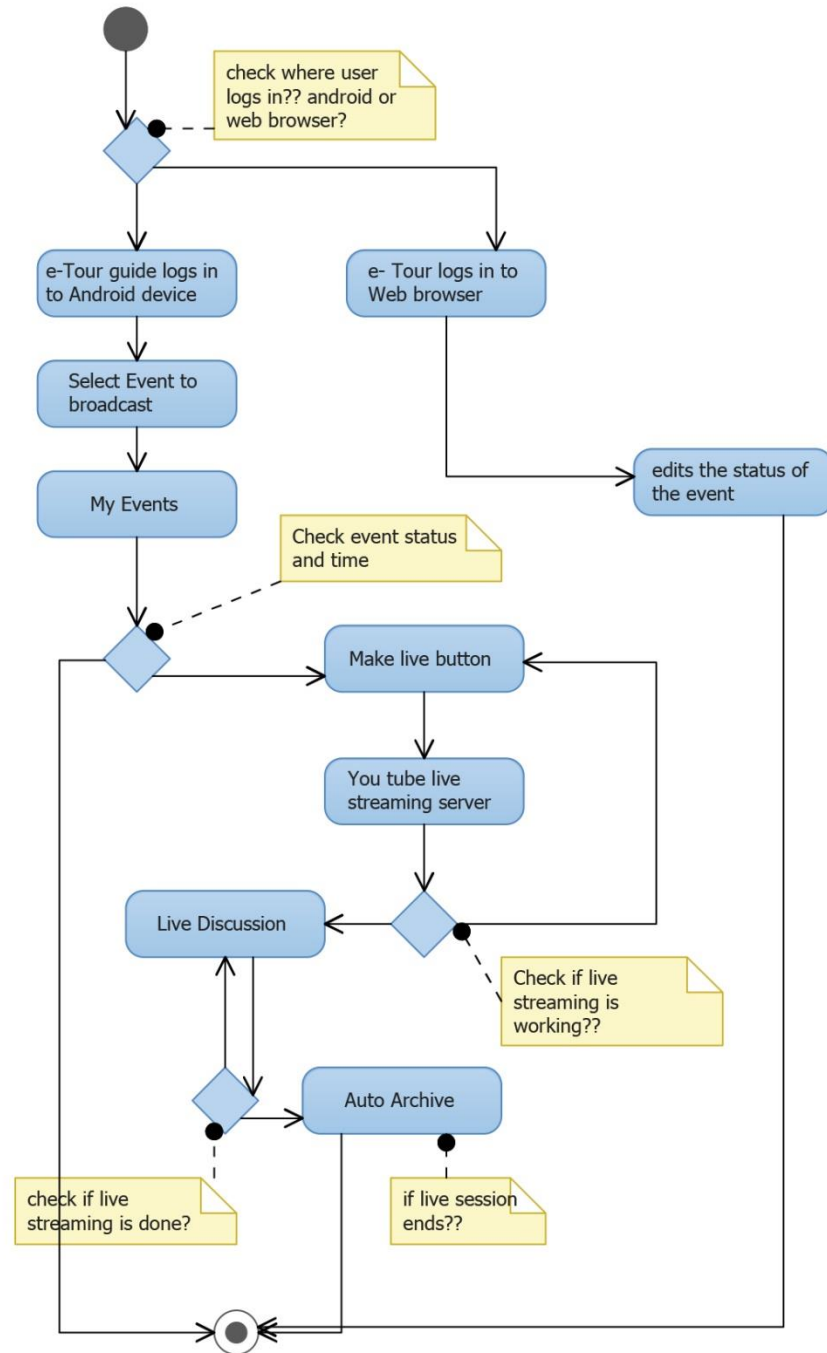


Figure 12: Activity Diagram for Broadcast Live event

Table 7: UCD for Watch Live Event

Use Case Name	Watch Live Event
Use Case No	UC05
Description	A user wanted to watch a live event in Thydash.
Participating Actors	e-Tourist
Entry Condition	<ol style="list-style-type: none"> 1. e- Tourist has logged in to his/her Thydash account. 2. e- Tourist joined the event.
Flow of Events	<ol style="list-style-type: none"> 1. e-Tourist selects event from upcoming events under My Tour. 2. The system connects to the YouTube live streaming server and e-Tourist starts watching the live event. 3. e-Tourist asks questions in the Q&A (Discussion).
Exit Condition	<ol style="list-style-type: none"> 1. User has successfully watched the event and the live session ended.
Alternatives	<ol style="list-style-type: none"> 1. The user clicks in the event early before the session started. <ul style="list-style-type: none"> • User will be forwarded to the event description and can see the status of the event. • User might leave the event too, and if it was a paid event, the payment handler will automatically handle the payment issue. 2. User unable to connect with live event. <ul style="list-style-type: none"> • The system will repeatedly try to connect to the session of the event.

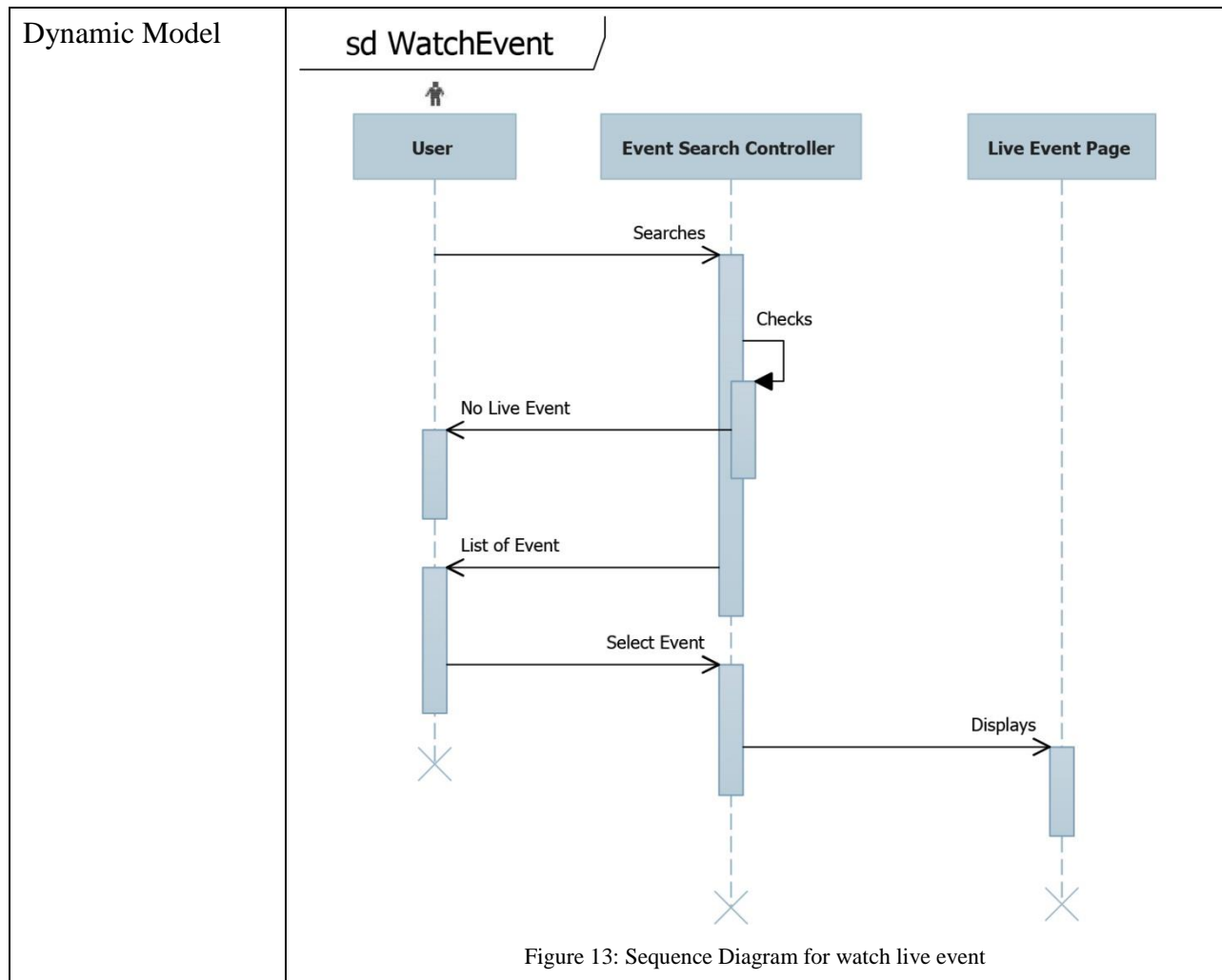
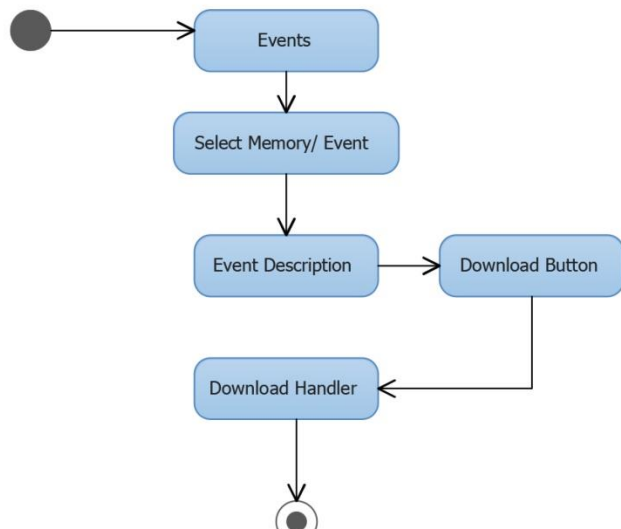


Table 8: UCD for download past events

Use Case Name	Download past Events
Use Case No	UC06
Description	A user wanted a video of past event he had joined before. Notice , User can also download events prepared by him/herself from his/her memories.
Participating Actors	e-Tourist, e-Tour guide
Entry Condition	1. e-Tourist/e-Tour guide logged in to his/her Thydash account
Flow of Events	1. e-Tourist/ e-Tour guide selects the specific past event he/she is interested in to download from the events page of my memories or my event. 2. clicks in the download button.

	3. Thydash will start loading the video and download manager will be invoked automatically.
Exit Condition	1. e-Tourist/e-Tour guide downloads the past event he wanted.
Dynamic Model	<p>act DownloadPastEvent.thydash</p>  <pre> graph TD Start(()) --> Events[Events] Events --> Select[Select Memory/ Event] Select --> Description[Event Description] Description --> Button[Download Button] Button --> Handler[Download Handler] Handler --> End((())) </pre> <p>Figure 14: State Diagram for Download past events</p>

3.5 Object Model

In this section we have described thydash in terms of models of the system using data dictionary, class diagram and object diagrams.

3.5.1 Data Dictionary

Here we will describe the data dictionaries¹⁹ for the classes sketched in our class diagram.

Table 9: User Data Dictionary

Variable	Variable name	Variable type	Variable width
User id number	Id	Numeric	11
First name of user	FirstName	Character	50
Last name of user	LastName	Character	50
Email of user	Email	Character	255
Password of user	Password	Character	50
Username of user	Username	Character	50
User account creation date	RegistrationDate	YY/MM/DD	
User's birth date	BirthDate	YY/MM/DD	
User's phone number	Phone	Numeric	14

Table 10: Event Data Dictionary

Variable	Variable name	Variable Type	Variable width
Event id number	Id	Numeric	11
Event name	Name	Character	50
Event Location	Location	Character	50
Description of Event	Description	Character	255
Promo Image location for an event	Image	Character	255
Amount to be paid for an event	Payment	Numeric	11

¹⁹ A **Data Dictionary** is a description the basic organization of a database. Typically a data dictionary will contain a list of variables in the database as well as the assigned variable names and a description of each type of variable (e.g. character, numeric, dates).

Event category	Category	1	1 = cultural 2 = social 3 = religious 4 = natural events
Event start date	FromDate	YY/MM/DD	
Event end date	ToDate	YY/MM/DD	
Status of event	Status	Character	50

Table 11: Video Data Dictionary

Variable	Variable name	Variable type	Variable width
Video id number	Id	Numeric	11
Video name	Name	Character	50
Video location	Location	Character	50
Video size	Size	Numeric	11
Video created date	DateCreated	YY/MM/DD	

Table 12: Payment Data Dictionary

Variable	Variable name	Variable type	Variable width
Payment id number	Id	Numeric	11
Event paid for	EventId	Numeric	11
User paid for the event	UserId	Numeric	22
Payment Date	PaymentDate	YY/MM/DD	
Payment Type	PaymentType	Character	50
Payment status	Status	Character	50

Table 13: Discussion Data Dictionary

Variable	Variable name	Variable type	Variable width
Discussion id number	Id	Numeric	11
User participating	UserId	Numeric	11
Event discussed on	EventId	Numeric	11

Question and answer	QuestionAndAnswer	Character	255
---------------------	-------------------	-----------	-----

Table 14: Rate Data Dictionary

Variable	Variable name	Variable type	Variable width
Rating id number	Id	Numeric	11
User rating	UserId	Numeric	11
Event being rated	EventId	Numeric	11
Rating of events	Rating	Numeric	11

Table 15: Comment Data Dictionary

Variable	Variable name	Variable type	Variable width
Comment id number	Id	Numeric	11
User commenting	UserId	Numeric	11
Event being commented on	EventId	Numeric	11
Comment	Comment	Character	255

Table 16: Group Data Dictionary

Variable	Variable name	Variable type	Variable width
group id number	Id	Numeric	11
Name of the group	UserId	Numeric	11
Group created date	DateCreated	YY/MM/DD	11
Users on the group	UserId	Numeric	11

3.5.2 Class Diagram

In this section we have prepared class diagram of our system. In addition to the features found in our scope, classes which will be used for optional features (Notification, Groups, etc.) are also included in the diagram.

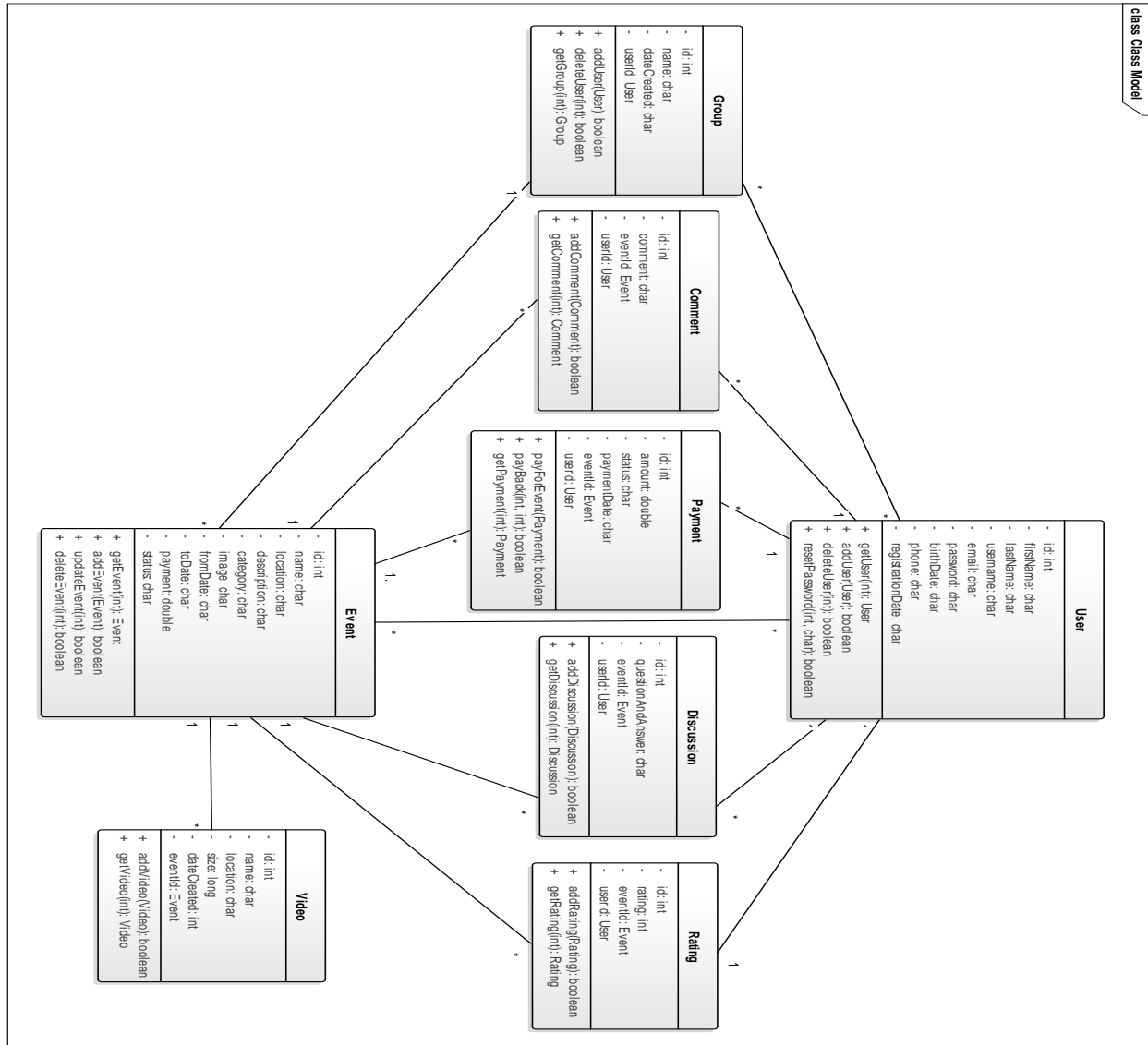


Figure 15: Class Diagram

3.5.3 Object Diagrams

In this section, we have prepared object diagrams²⁰ for some functionalities of the system.

Object Diagram for commenting on an event

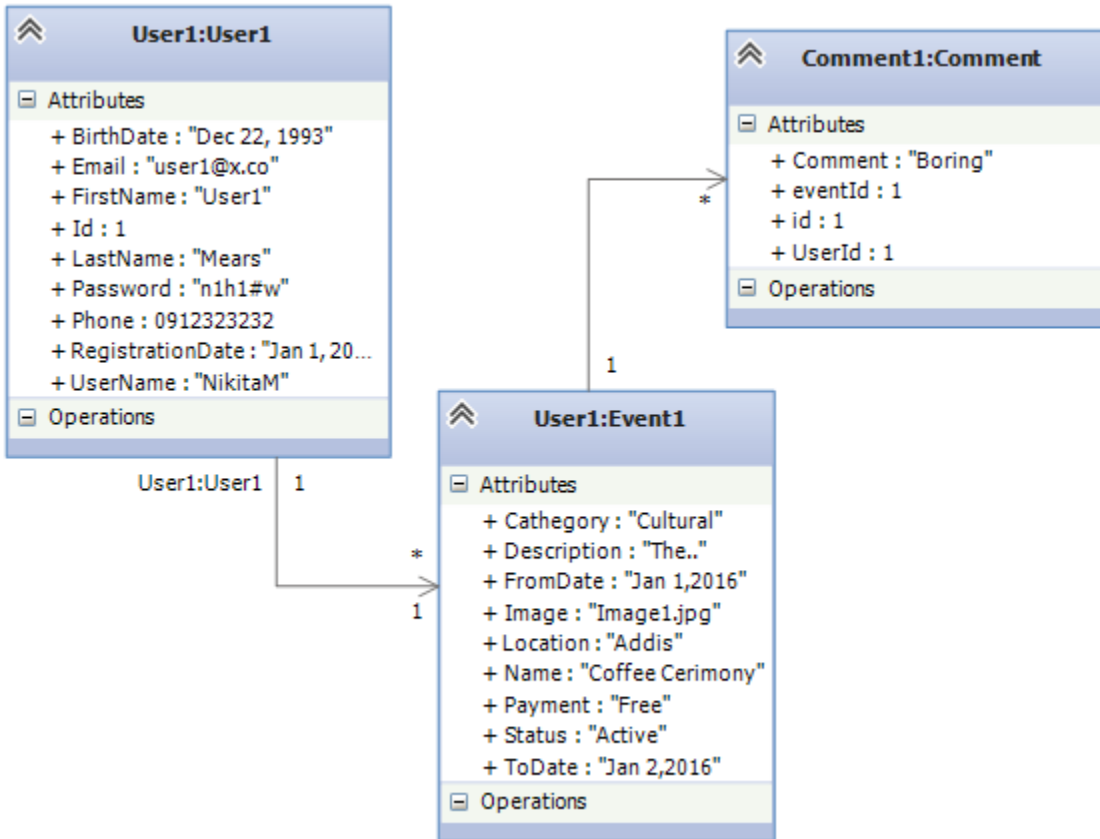


Figure 16: Object Diagram for Commenting on an event

²⁰ An **object diagram** in the Unified Modeling Language (UML) is a **diagram** that shows a complete or partial view of the structure of a modeled system at a specific time.

Object Diagram for discussion

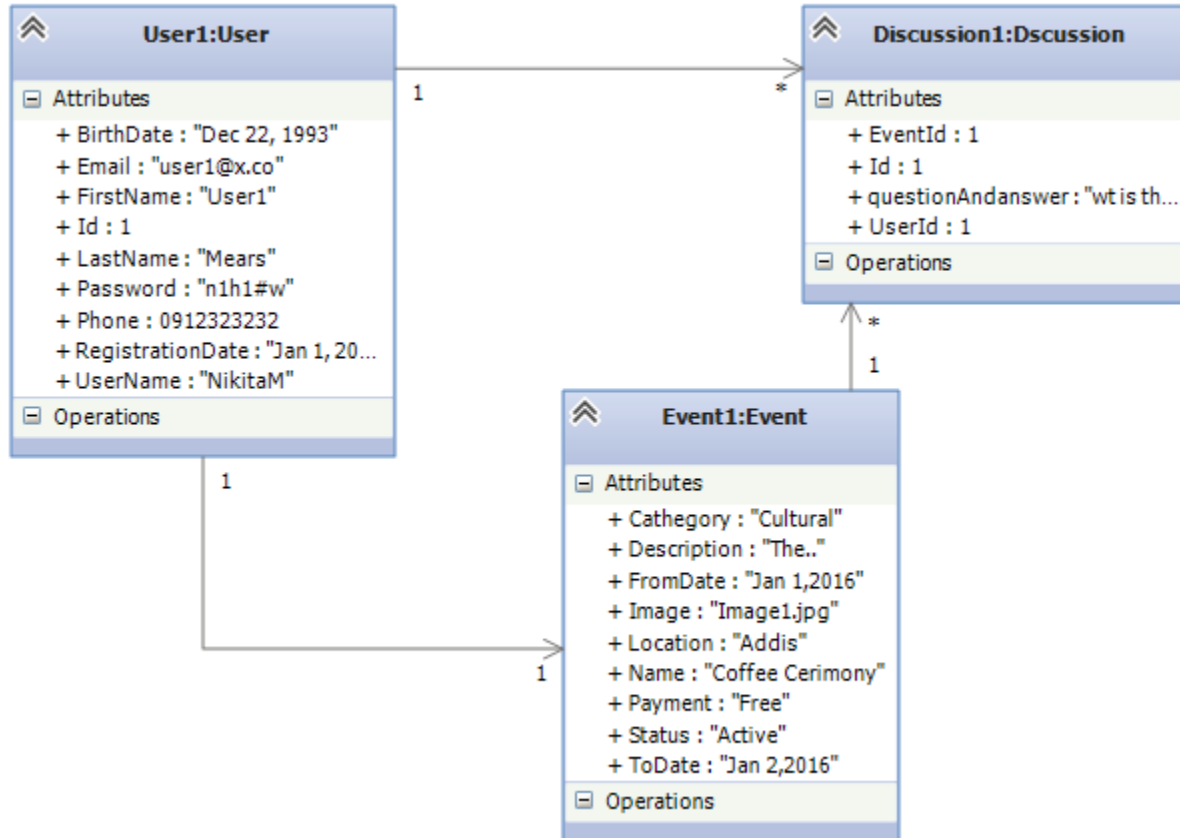


Figure 17: Object Diagram for discussion

Object Diagram for creating an event

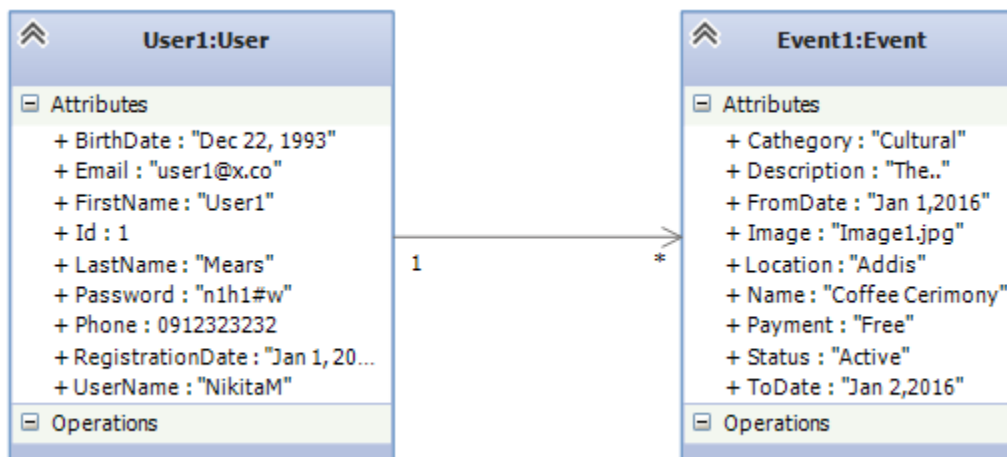


Figure 18: Object Diagram for creating an event

Object Diagram for joining an event

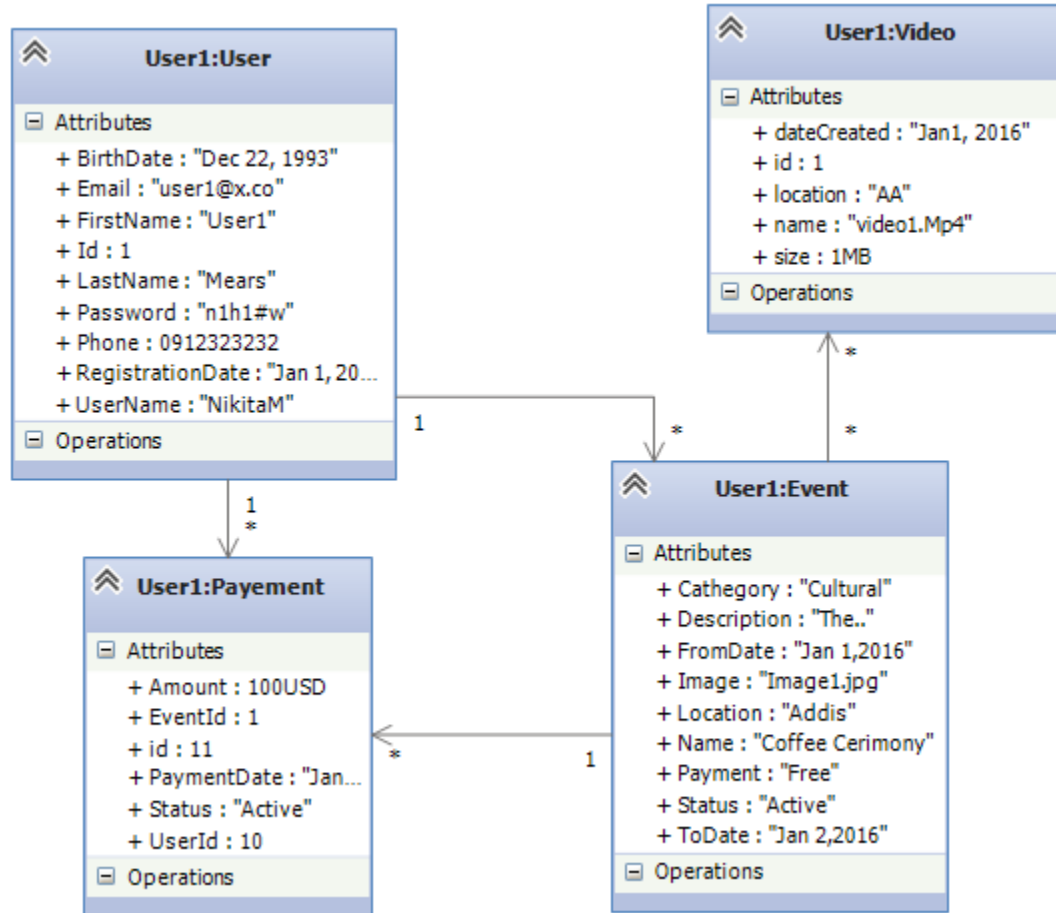


Figure 19: Object Diagram for joining an event

Chapter 4: System Design

“There are two ways of constructing a software design. One way is to make it so simple that there are obviously no deficiencies. And the other way is to make it so complicated that there are no obvious deficiencies.”

-- C.A.R. Hoare²¹

4.1 Overview

The main purpose of this chapter is to provide a brief overview of the software architecture and the design goals. In this section we will discuss and describe design goals set by the project, subsystem decomposition , hardware/software mapping (with UML deployment diagrams), data management, access control using access matrix, control flow mechanisms, and boundary conditions using use case description and diagram.

4.2 Design Goals

Our goals when designing thydash are:

Usability: thydash should be simple and easy to use for an average user. A user should be able to effectively use thydash without prior experience and without professional support.

Security: thydash should be a secure platform. i.e unauthorized people shouldn’t be able to maliciously access thydash.

Maintainability: thydash should be designed in a way that it can be easily maintained i.e components in the design should have high cohesion and less coupling.

Performance

- **Availability:** thydash should be available every time except for maintenance times.
- **Response time:** thydash should be well interactive to satisfy users of the system.

Interface: thydash should meet the interface requirements specified by YouTube Live API and Payment APIs, such as BrainTree.

²¹ C. A. R. Hoare is a British computer scientist. He developed the sorting algorithm quicksort in 1959/1960.

4.3 Proposed System Architecture

4.3.1 High Level Design

In thydash we will use a **three tiered** client-server architecture where clients, database and web servers are physically separated.

- **Android Devices:** the mobile application is deployed in Android devices; Android Digital Cameras and Android smart phones
- **Web Clients:** the web version of thydash is accessed from different browsers installed in devices with platforms such as MAC, windows and Linux.
- **Database and Web Servers:** both the database and the Django web-servers will be deployed in the same machine. The web server will run contain services for user management, event management and live streaming. These services will interact with our database servers.

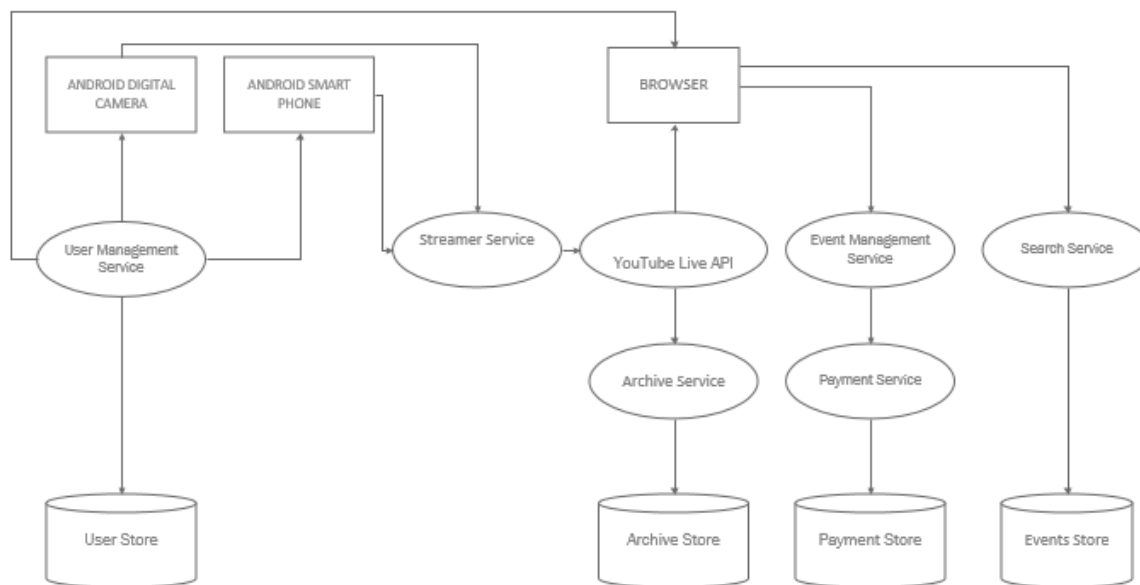


Figure 20: High Level Architecture of thydash

Notes:

- The streamer function accepts real-time videos from android powered smart phones or cameras and redirects the video to the YouTube Live streaming API.
- YouTube Live streaming API will transfer live video to browser clients.
- Archive service accepts the video transmitted from YouTube Live streaming API will store them to the Archive store.

- Payment handler service, implemented using PayPal and/or other payment systems, handles payments for an event from the browser client and stores them to payments store.
- The search service gets search criteria from browser clients queries the Events Store and responds results to the browser.
- In implementing Thydash we'll use MySQL as our database server, C and Java in android application development and Django Web Framework for web application development.

To implement the web version of thydash we will use **Django** Web Framework i.e. we will use a **Model-Template-View** architectural style.

- **URL Dispatcher:** maps the requested URL to a view function and calls it.
- **View:** performs the requested action, which typically involves reading or writing to the database. It may include other tasks, as well.
- **Model:** defines the data in Python and interacts with it. Our data will be stored in MySQL.
- **Template:** holds the presentation logic of the system. It will return HTML pages.

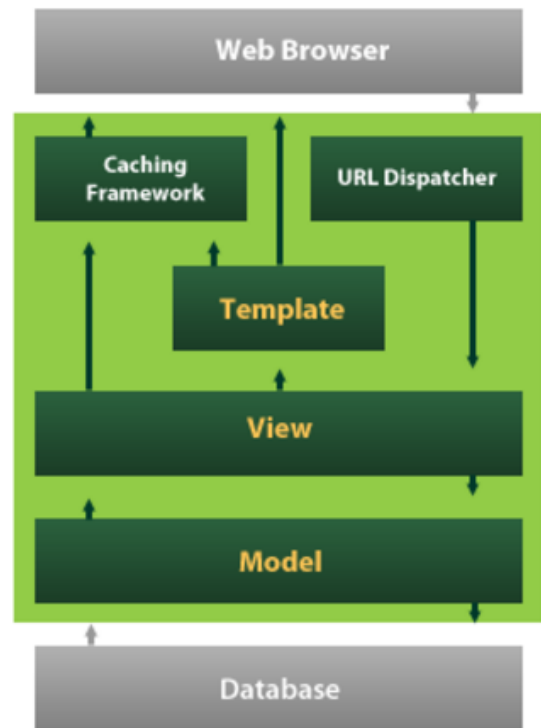


Figure 21: Django MTV Architecture

The web version of thydash will expose **REST API** endpoints about users and events. These endpoints will be used in our android application to authenticate users and to get information about events.

To implement the android version of thydash, we will use data and view layers. The data layer will have database helper which interacts with SQLite database, preferences helper which interacts with shared preferences, thydash API service which interacts with the REST API end point of thydash, YouTube API Service which communicates with YouTube Live API.

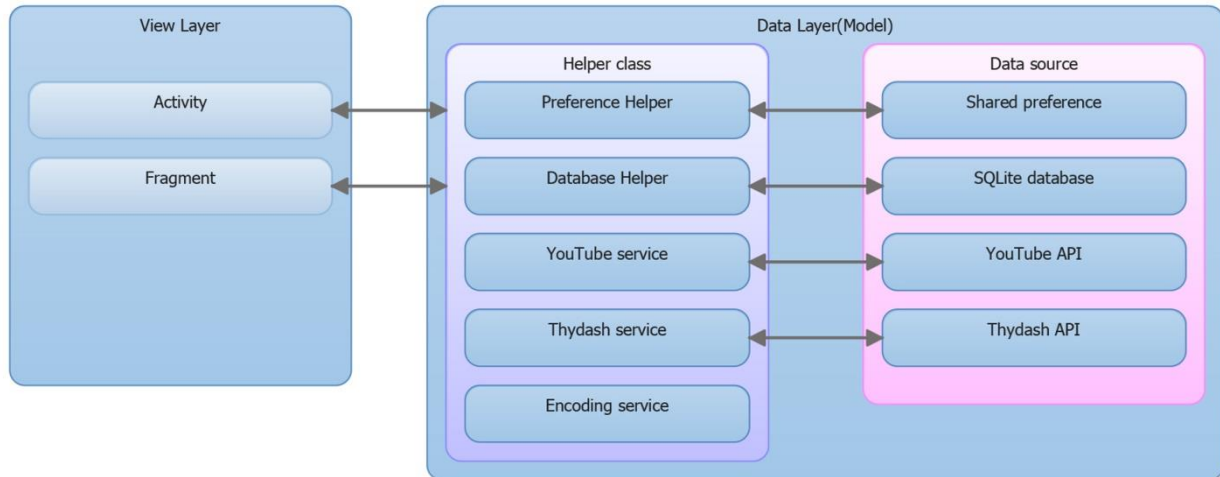


Figure 22: thydash Android application Architecture

4.3.2 Subsystem Decomposition

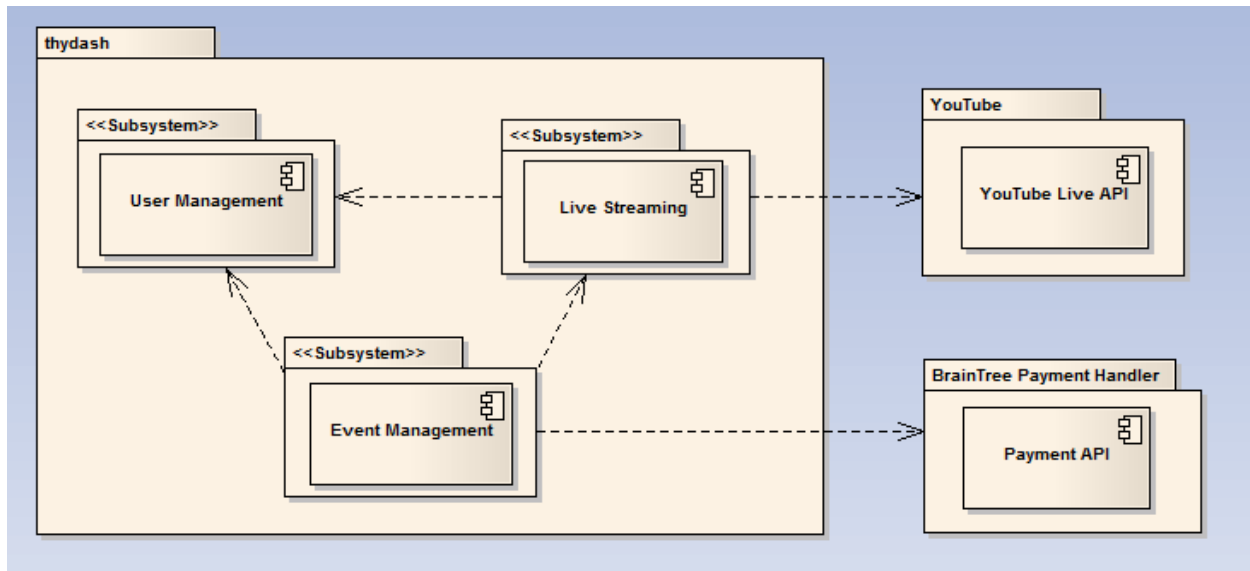


Figure 23: Subsystem Decomposition of thydash

Thydash will have mainly 3 subsystems. These subsystems are listed and described as follows:

- The **User Management subsystem** will take care of all issues related to the user i.e. login, create account, user profile management, etc. and it will interact with event management subsystem by giving information about the current user.
- The **Event Management subsystem** will handle issues related to events i.e. create event, delete event, join event, leave event, watch event, etc. It interacts with user management subsystem to get user information and with live streaming subsystem to get live streamed

videos of an event. In addition, it is integrated with payment handler as external subsystem.

- The **Live Streaming subsystem** will handle video encoding, decoding, and video streaming. It will be integrated with the YouTube Live streaming API as an external subsystem.

4.3.3 Hardware / Software Mapping

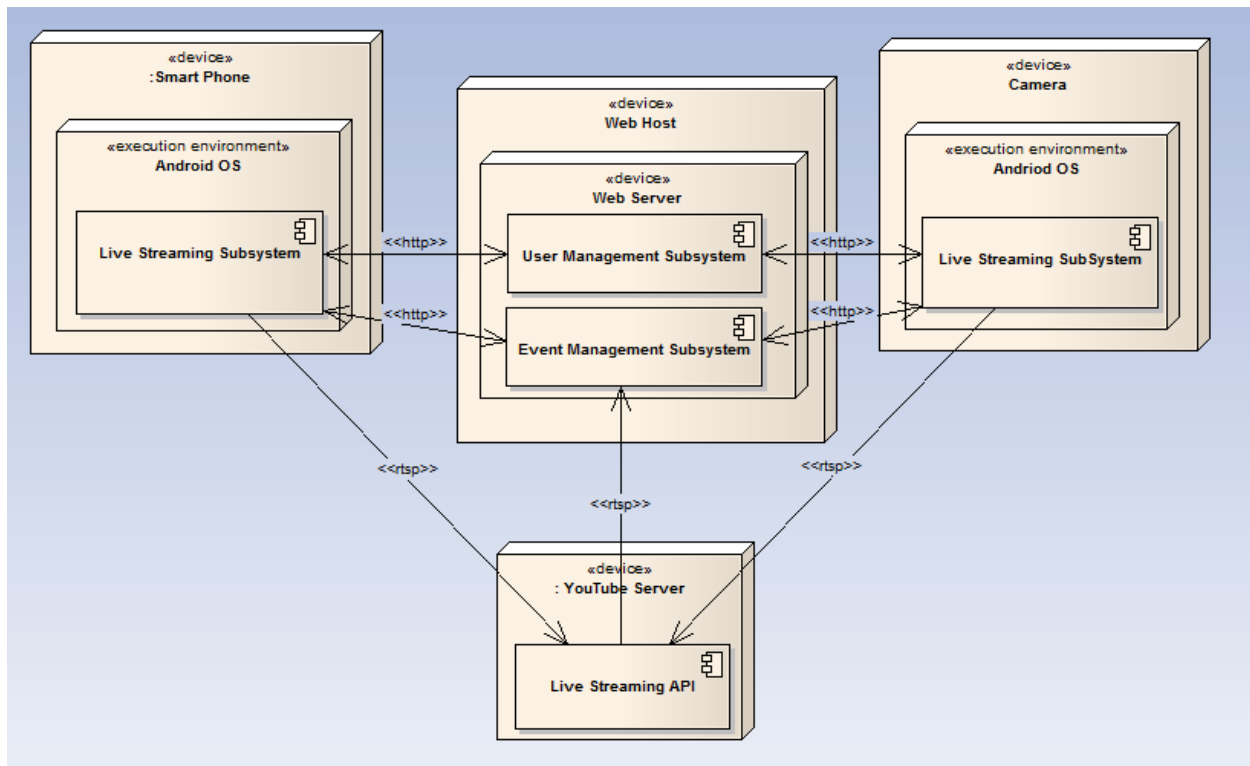


Figure 24: Hardware Software Mapping

In this section, we have used hardware/software mapping to indicate which subsystems of thydash are deployed in which hardware components.

- Live Streaming sub system is deployed/ installed on android devices i.e camera or smartphone.
- We will use YouTube live streaming API which is found in YouTube server.
- User and Event Management Subsystems are deployed on apache webserver with WSGI plugin.

4.3.4 Persistent Data Management

User related Information, Event related Information, Payment related Information are the persistent objects that we have identified. We are going to use Relational Database to store those. The type of relational database that we are going to use is, MySQL. Image and Video data will be stored as flat files.

4.3.5 Component Diagram

In this section we used component diagrams describe the organization of the components thydash. These components are organized in a way to meet the requirements of the system.

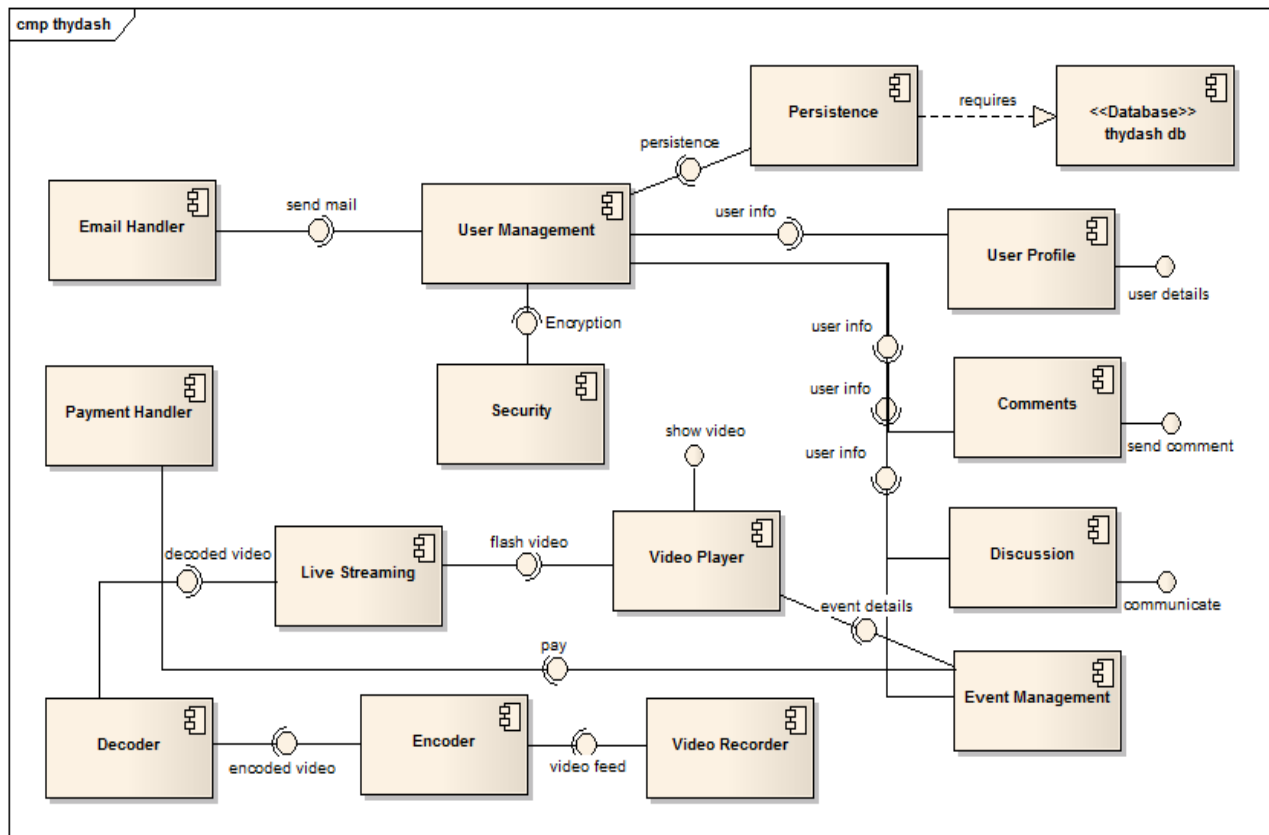


Figure 25: Component Diagram

4.3.6 Deployment Diagram

We are using this diagram (deployment) to depict the relationship among runtime components and nodes. In this context, components are self-contained entities that provide service to other components or actors and a node is a physical device or an execution environment in which components are executed. Nodes are represented by boxes containing component icons and Communication paths between nodes are represented by solid lines and the protocol is described with stereotype in the communication path.

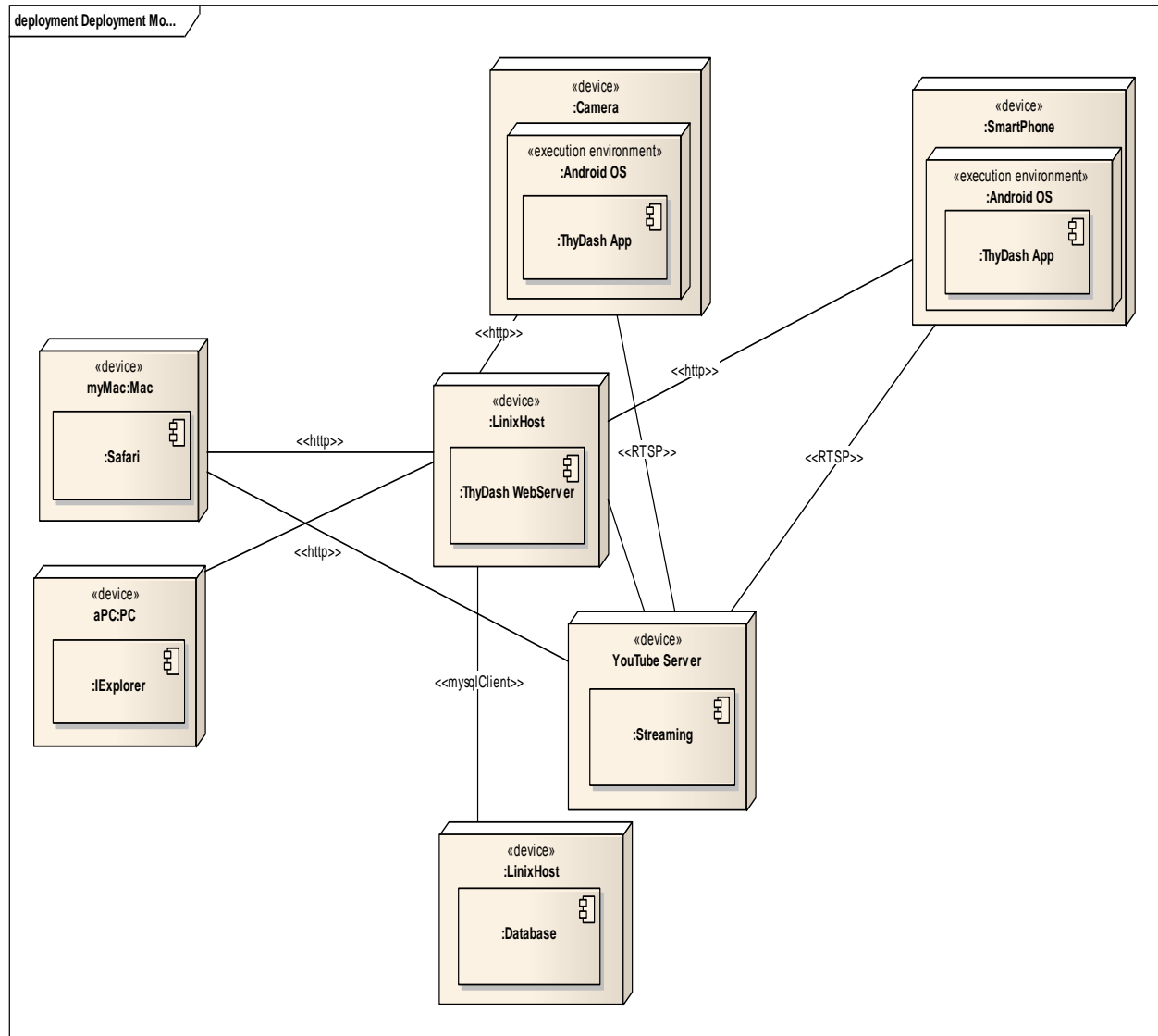


Figure 26: Deployment Diagram

4.3.7 Boundary Conditions

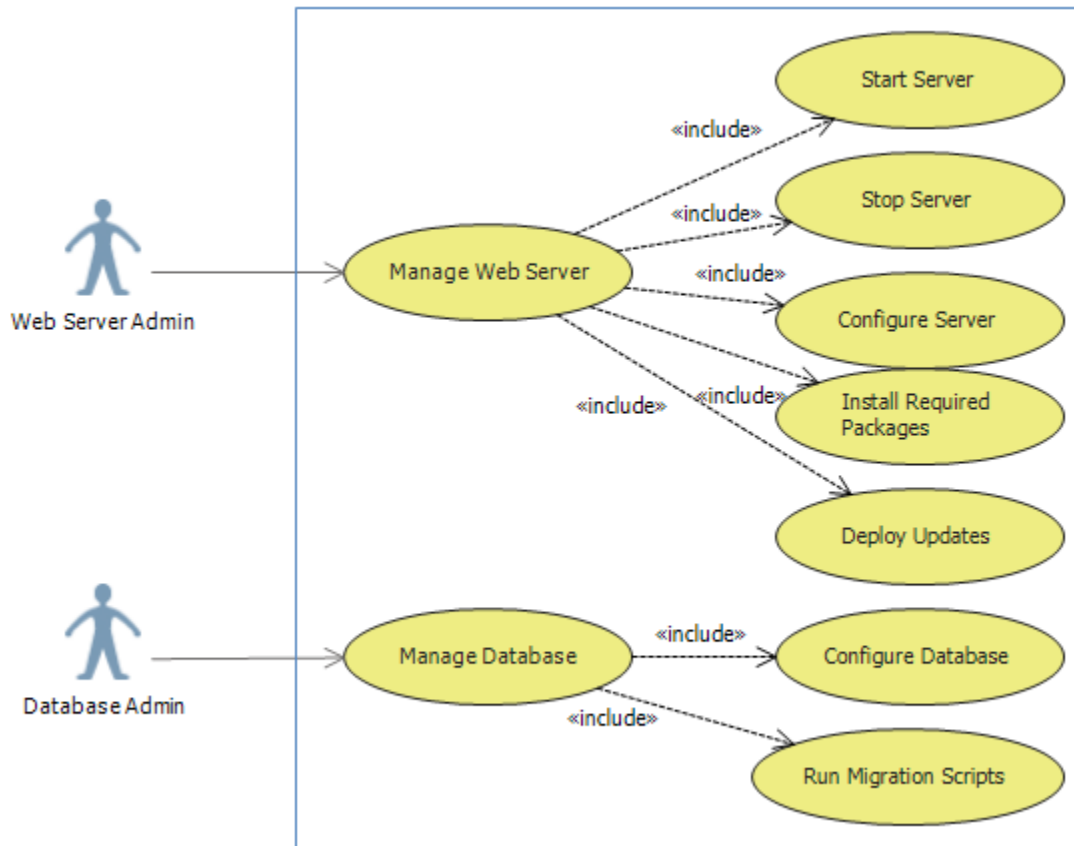


Figure 27: UseCase Diagram for Boundary Conditions

Table 17: UCD for Start Server

Use Case Name	Start Server
Description	Starting thydash web server
Participating Actors	Web Server Admin
Entry Condition	1. The web server admin has logged into the server machine.
Flow of Events	2. Upon successful login, the web server admin executes run server command. 3. If there are uninstalled packages and libraries and if there are problems in database and other configurations, the server notifies the web server admin.
Exit Condition	4. Thydash web server is up and running.

Table 18: UCD for Stop Server

Use Case Name	Stop Server
Description	Stopping thydash web server
Participating Actors	Web Server Admin
Entry Condition	1. The web server admin has logged into the server machine.
Flow of Events	2. Upon successful login, the web server admin executes stop server command.
Exit Condition	3. Thydash web server is stopped.

Table 19: UCD for Configure Server

Use Case Name	Configure Server
Description	Adding different configurations to thydash web server
Participating Actors	Web Server Admin
Entry Condition	1. The web server admin has logged into the server machine.
Flow of Events	2. Upon successful login, the web server admin executes configuration command. 3. The system gets new configuration i.e database configuration, security configuration, API Keys configuration, etc.
Exit Condition	4. Thydash web server is using new configurations.

Table 20: UCD for Install Required Packages

Use Case Name	Install Required Packages
Description	Installing required packages and libraries.
Participating Actors	Web Server Admin
Entry Condition	1. The web server admin has logged into the server machine.
Flow of Events	2. Upon successful login, the web server admin executes package installation command. 3. The system installs the package and its requirements.
Exit Condition	4. Thydash web server is able to use the new package.

Table 21: UCD for Deploy Updates

Use Case Name	Deploy Updates
Description	Installing required packages and libraries.
Participating Actors	Web Server Admin
Entry Condition	1. The web server admin has logged into the server machine.
Flow of Events	2. Upon successful login, the web server admin deploys new version of thydash. 3. If there are required packages to the newly updated system, the web server notifies it. Otherwise, the system will continue working.
Exit Condition	4. Thydash web server is updated with next version of the system.

Table 22: UCD for Configure Database

Use Case Name	Configure Database
Description	Adding new database configurations
Participating Actors	Database Admin
Entry Condition	1. The database admin has logged into the database server machine.
Flow of Events	2. Upon successful login, the database admin configures thydash database.
Exit Condition	3. Thydash database starts using the new configuration.

Table 23: UCD for Run Migration Scripts

Use Case Name	Run Migration Scripts
Description	Changing structure of database by running migration scripts
Participating Actors	Database Admin
Entry Condition	1. The database admin has logged into the database server machine.
Flow of Events	2. Upon successful login, the database administrator runs migration scripts. 3. If the database has relationship constraints that bind the new database structure from being enabled, the database gets notified.
Exit Condition	4. Thydash database gets new database structure.

4.3.8 Database Design

Since we'll use the **ORM**²² (object relational mapping) feature of Django Web Framework, we aren't supposed to create database design. The ORM feature automatically maps the classes we have created into relational database tables.

In this section, we have included snapshot of the database design that is generated by the Django ORM. But, for the sake of readability, we have minimized contents of some tables.

²² **Object Relational Mapping (ORM)** is used by Django web framework to map python classes to relational database tables.

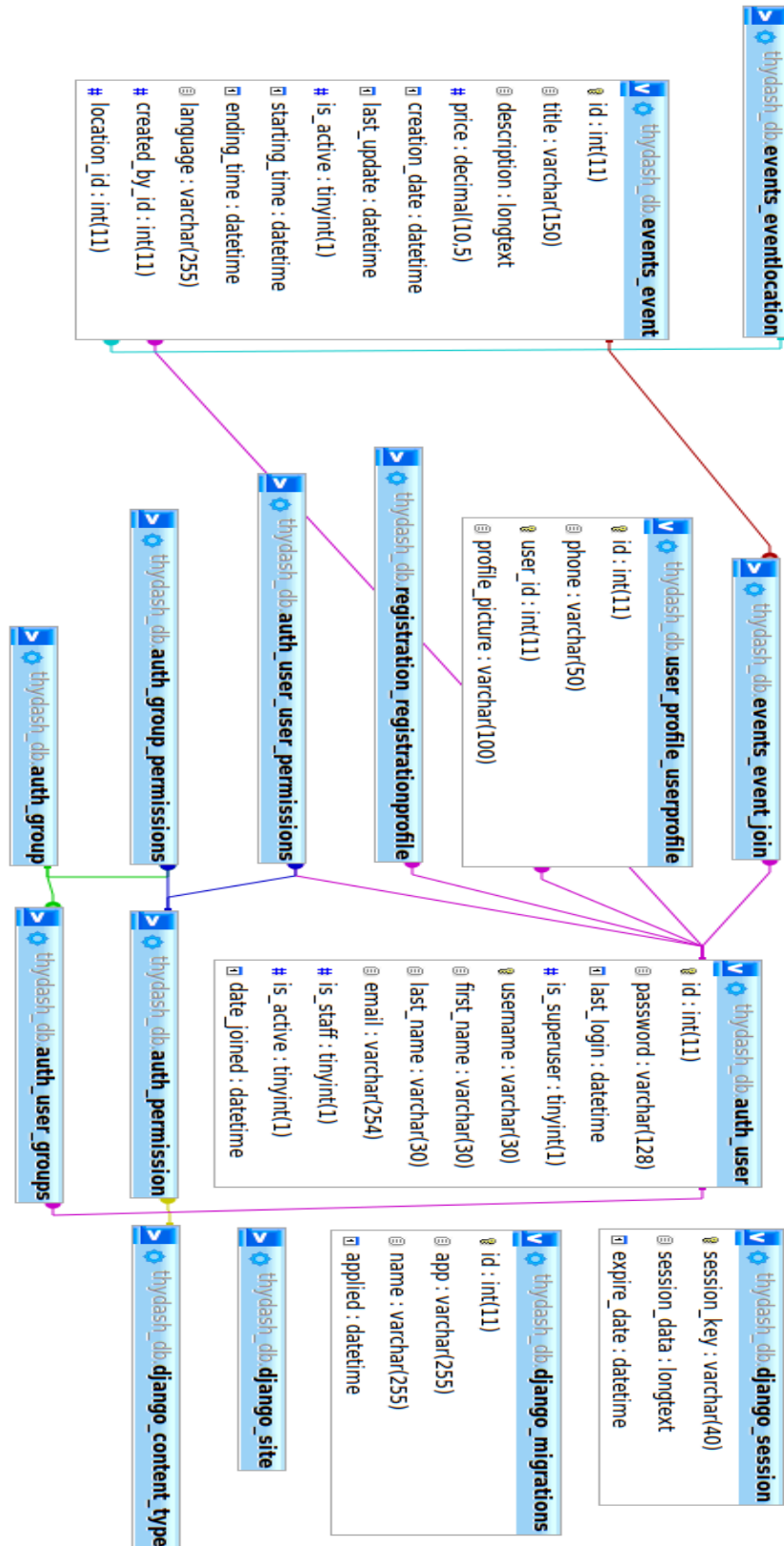


Figure 28: Database Design

4.3.9 Access Controlling

Our product is a commercial web platform; it only supports users with similar privilege. The only access control that we are going to apply is for guest users until they register themselves. We have prepared a table showing what functionalities guest and registered users can access.

Table 24: Thydash Access Controlling

User	Functionalities User can access
Guest User	<ul style="list-style-type: none">✓ Create Account✓ Search Event✓ Check Event Description
Registered User	<ul style="list-style-type: none">✓ Log In✓ Edit Profile✓ Search Event✓ Join Event✓ Watch Live Event✓ Create Event✓ Broadcast Live Event✓ Watch Archive videos✓ Download Archive videos

4.3.10 User Interface Design

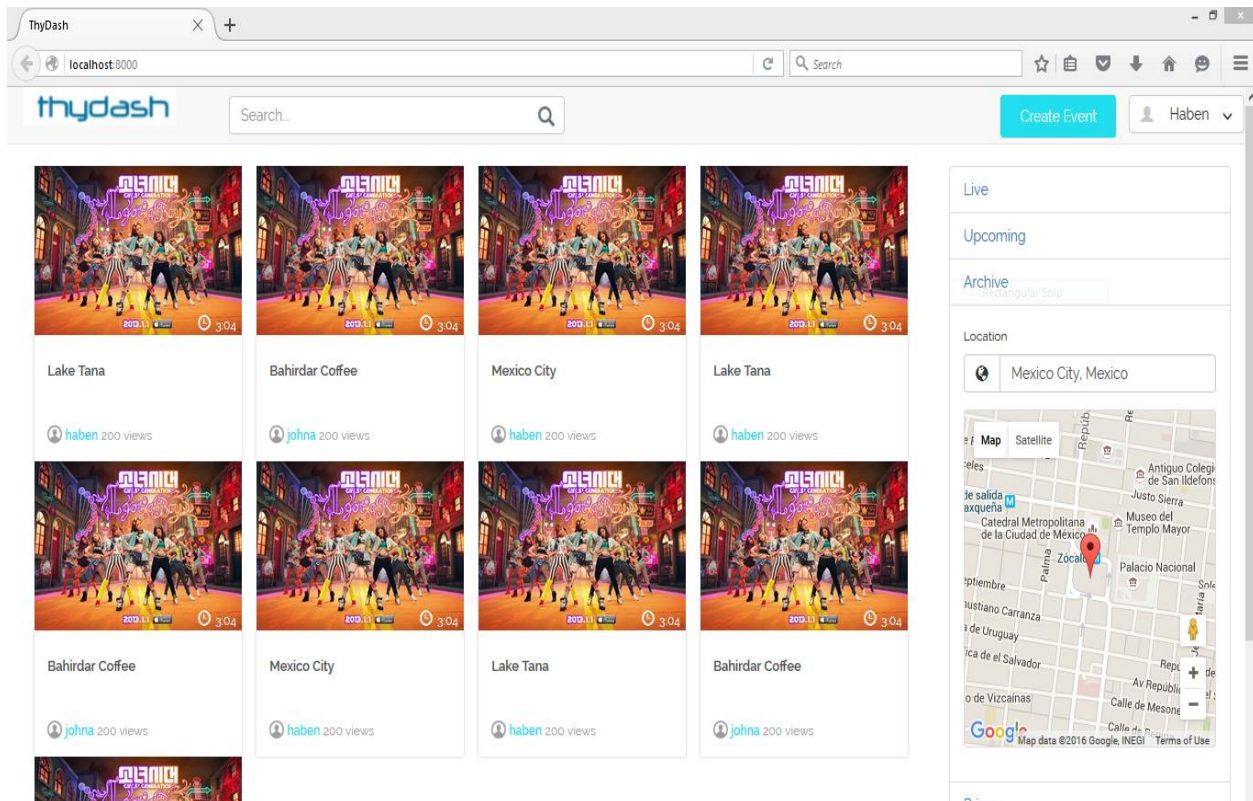


Figure 29: Web User Interface

Web User Interface

- This is logged in user home page.
- The logo is link to the home page of the thydash.
- The search box searches events by their title.
- The thumbnails on the left are list of events.
- Live, upcoming, archive and locations are filters for the search result.
- A user can create his events by clicking on create event button.
- Users can logout by clicking logout link under username dropdown menu.
- User can update his profile info by clicking the profile link under username dropdown menu.
- User can see the details of the event like event description, event location, event creator (host), payment if it is paid event by clicking on the event's thumbnail.
- If the user is not registered thydash user, the create event and the account option will not be available.

Android User Interface

- To use this android feature a user should have thydash id.
- This is the live stream page mockup screen.
- A user logs in with his thydash id.
- A user can start and stop streaming by using stop / start button.
- A user can see users question on his event by clicking on discussion button and answer to the questions the users who joined his event are asking.
- And last but not list the event creator sees what he is recording.

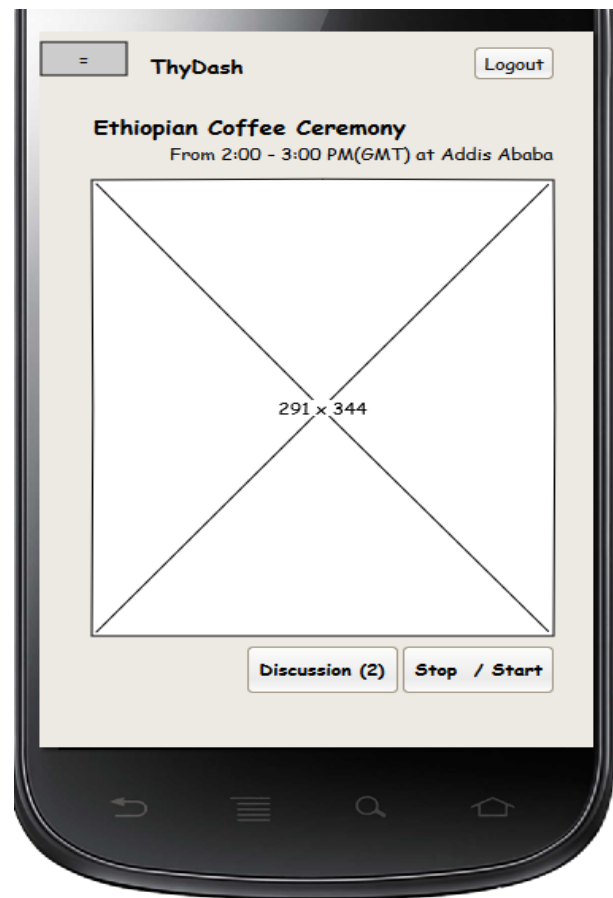


Figure 30: Android User Interface

Chapter 5: Features List

*“Attention to detail can't be (and never is) added later.
It's an entire development philosophy,
methodology, and culture”
Marko Arment²³*

5.1 User Stories

5.1.1 Android User Stories

Login with Google

As an E-Tour Guide

I want to login to my thydash account by using my Google Account

So that I can create live stream events, if they are live.

Logout

As an E-Tour Guide

I want to logout from my thydash account

So that I will be asked credentials the next time I access the app.

List of Events

As an E-Tour Guide

I want to see list of events I created

So that I can choose one of my events

Live stream

As an E-Tour Guide

I want to live stream my event

So that e-Tourists who joined my event can watch live.

Discussion

As an E-Tour Guide

I want to discuss about my live event

So that I can answer questions that e-Tourists who joined my event have asked.

5.1.2 Web User Stories

Search

As a Guest User, E-Tourist, E-Tour Guide

I want to search events by entering keywords

So that I can get events which have contents similar to the keywords that I have entered.

²³ **Marco Arment** is an American iOS and Web developer, as well as a technology writer and former magazine editor, living in Hastings-on-Hudson, New York. He is known for co-founding Tumblr, creating Instapaper, and Overcast.

Filter

As a Guest User, E-Tourist, E-Tour Guide

I want to filter events by specifying location, language of transmission, payment and other filters

So that I can get events which satisfy the filters I have applied

Sign Up with Google

As a Guest User

I want to create an account by using my Google account

So that I can have access to thydash services that registered users can access

Profile

As an E-Tourist, E-Tour Guide

I want to see my personal profile and update my personal info

So that I can provide info to fellow users

Login with Google

As an E-Tourist, E-Tour Guide

I want to login to my thydash account by using Google's authentication

So that I can access all thydash services

Home – Logged In

As an E-Tourist, E-Tour Guide

I want to see events that I joined and other popular events to join

So that I can have an overall know-how about the information that exists

Create Event

As an E-Tour Guide

I want to create an event by entering required details about it

So that I can have E-Tourists who will join my event and I can get money if I created a paid event.

Join Event

As an E-Tourist

I want join an event by paying the amount of money required, if needed.

So that I can participate in that live tour

List of People who joined event

As an E-Tour Guide

I want to see list of users who joined my event

So that I can know the number and demography of users who joined the event

Live Event

As an E-Tourist

I want to see currently live events that I have joined

So that I can take part in the live event

Discussion

As an E-Tour Guide, E-Tourist

I want to discuss about a live event

So that I can ask as an E-Tourist or answer questions those e-Tourists who joined my event have asked.

Upcoming events

As an E-Tourist

I want to see events that I joined which will be streamed in the future

So that I can watch live or update my status on the event

Memories

As an E-Tourist

I want to see events that I had participated in the past

So that I can memorize by watching those events and download them

Update Event Details

As an E-Tour Guide

I want to update details of events I created

So that a new set of information is applied to the event.

Logout

As an E-Tour Guide, E-Tourist

I want to logout from my thydash account

So that I will be asked credentials the next time I access the app.

Chapter 6: Features Implementation

"Organizations are successful because of good implementation, not good business plans."

Guy Kawasaki²⁴

6.1 User Management Module

6.1.1 Features

The features that will be handled by user management module are:

- Sign up with Google®.
- Login with Google®.
- Logout from thydash.
- Profile management.
- Authorization.

6.1.2 MVC Class Diagram

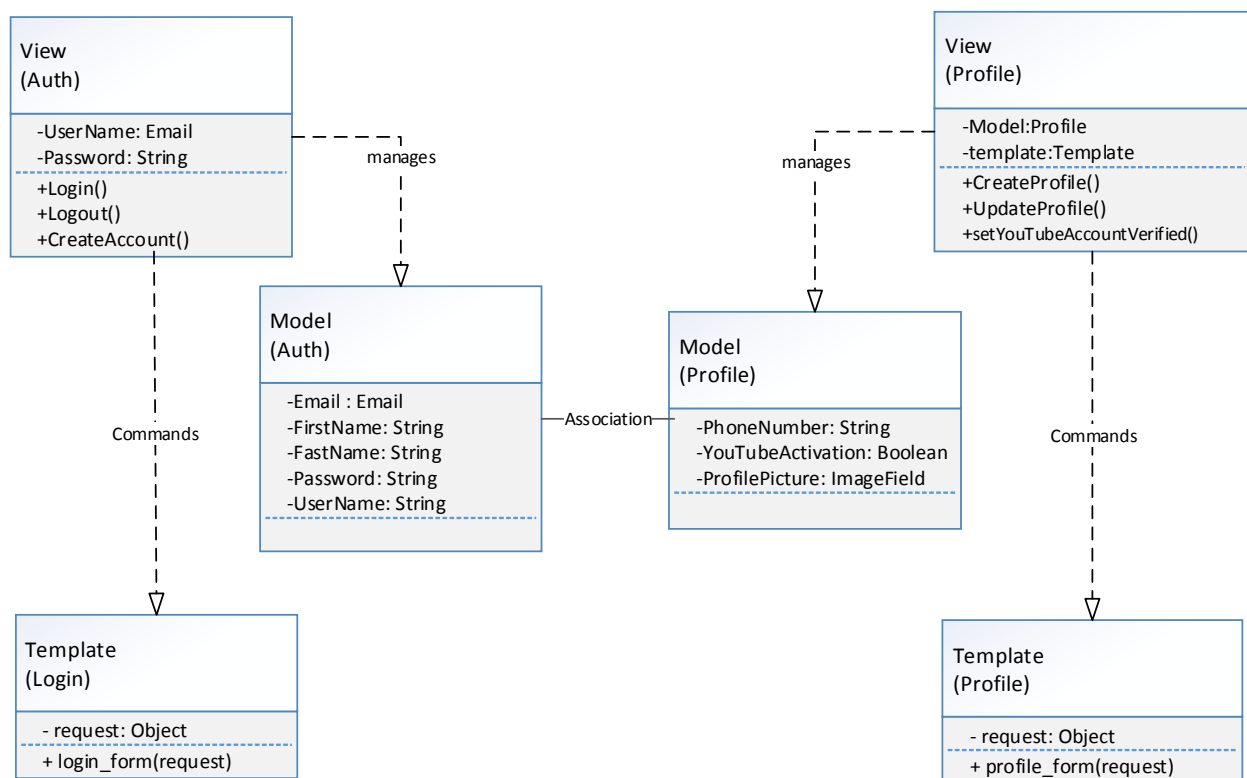


Figure 31 MVC class diagram for User Management

²⁴ **Guy Kawasaki** is an American marketing specialist, author, and Silicon Valley venture capitalist. He was one of the Apple employees originally responsible for marketing their Macintosh computer line in 1984.

6.1.3 Screenshots

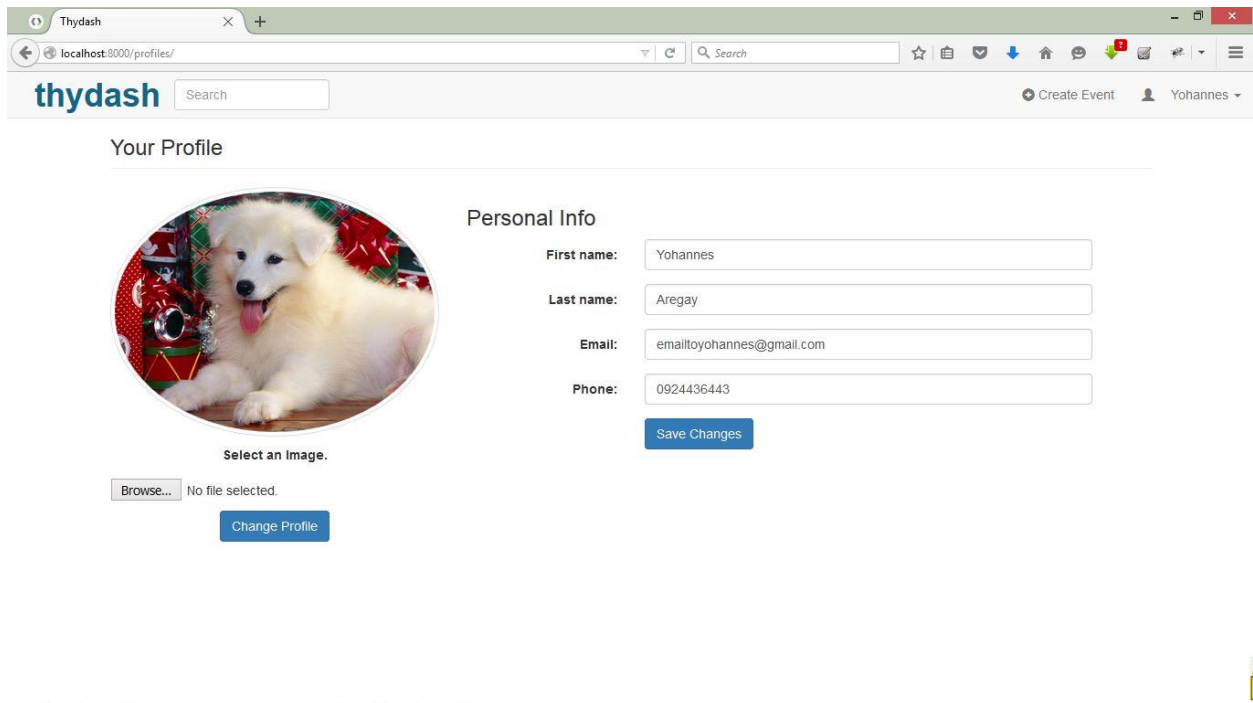


Figure 32 Screenshot for Profile Page

6.1.4 Tools and Technologies Used

The tools and technologies that we used to implement user management module are:

Tools

- PyCharm

Technologies

- Front end framework: *Twitter Bootstrap v3*
- JavaScript Library: *JQuery*
- Programming Language: *Python*
- Python Web framework: *Django*
- Django Libraries: *python-social-auth, Pillow*
- APIs: *Google Plus API*

6.1.5 Prototype Setup

To install and use thydash user management module:

- The client should use a JavaScript enabled web browser.

- The Django Application should be hosted on python 3 and Django 1.8.7 installed machine having WSGI (web server gateway interface) web server.

6.2 Event Management Module

6.2.1 Features

The features that will be handled by event management module are:

- Creating event
- Updating event
- Deleting event
- Joining event
- Handling payments
- Searching events
- Show list of events.
- Show Live events.

6.2.2 MVC Class Diagram

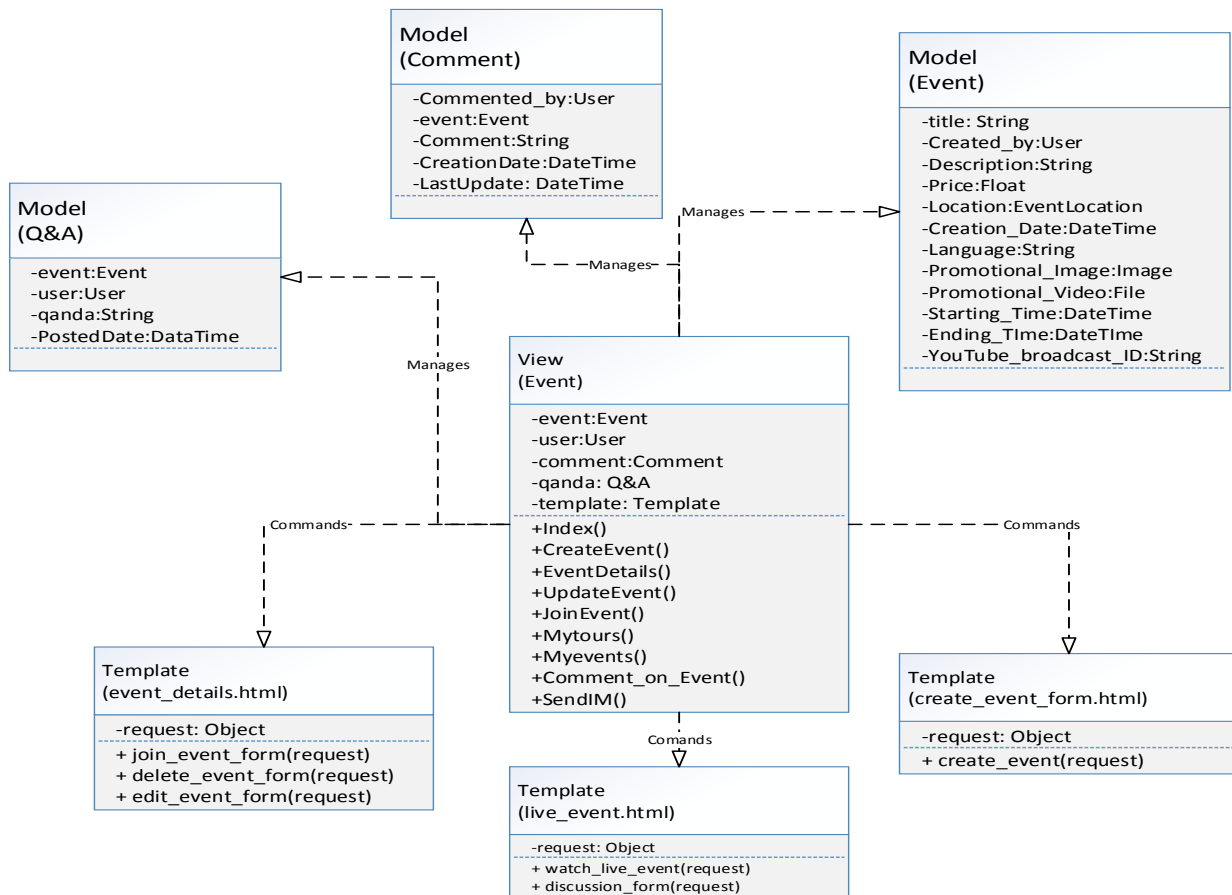


Figure 33 MTV model for Event Management

6.2.3 Screenshots

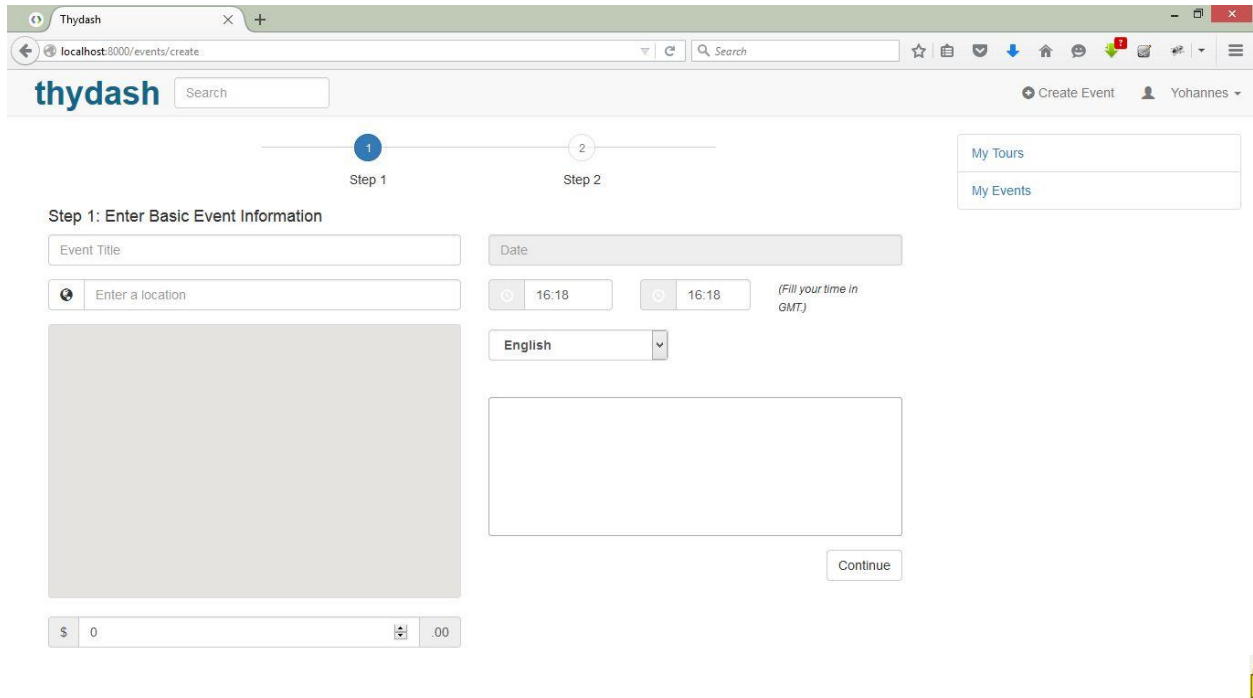


Figure 34 Screenshot for Create Event

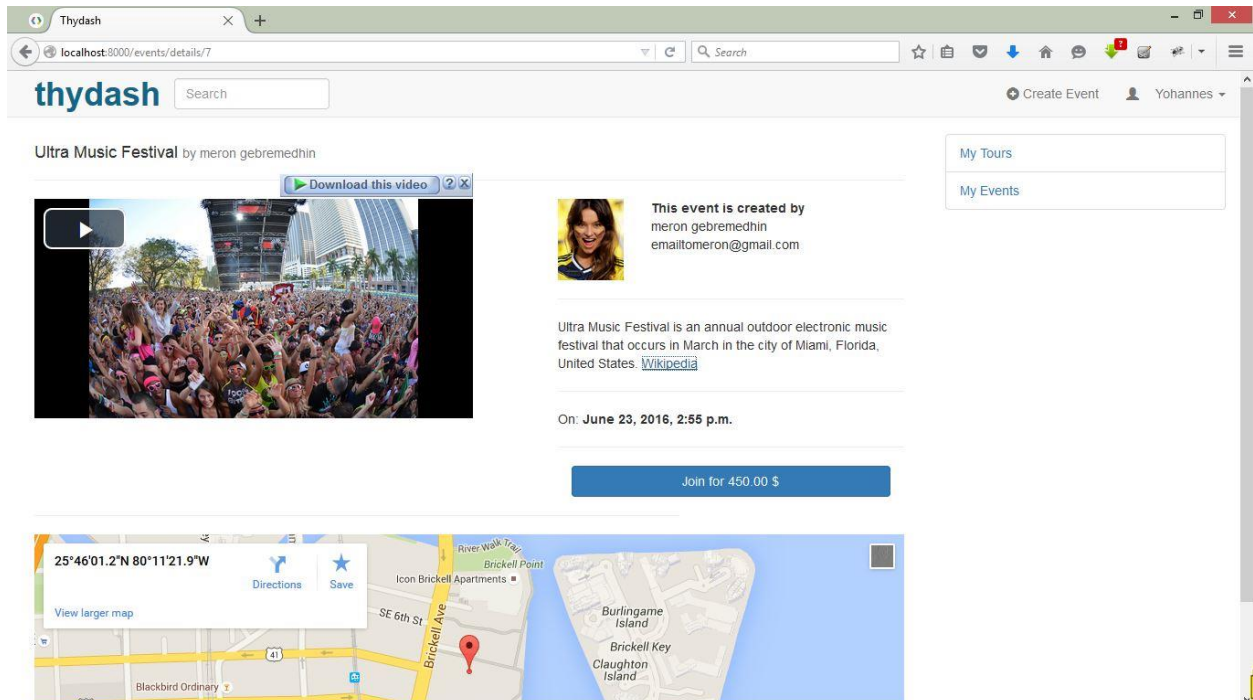


Figure 35 Screenshot for Event Details of Paid Event

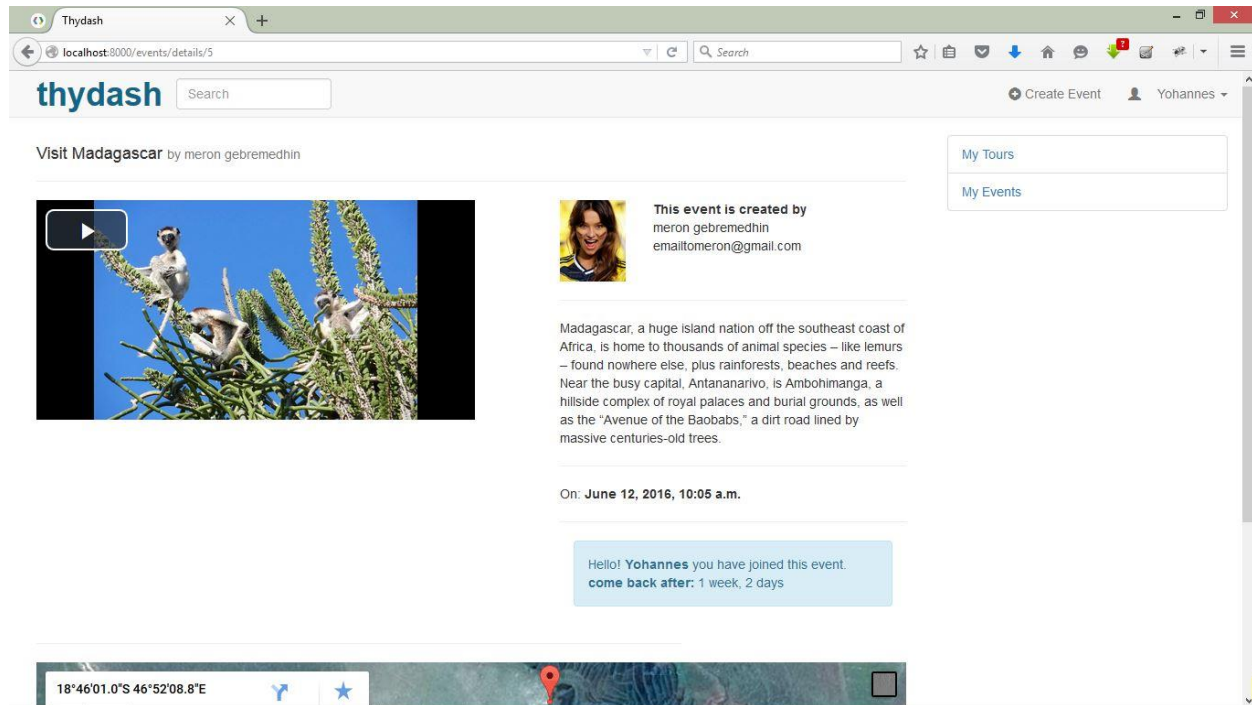


Figure 36 Screenshot for Event Detail of Paid Event

6.2.4 Tools and Technologies Used

The tools and technologies that we used to implement user management module are:

Tools

- PyCharm

Technologies

- Front end framework: *Twitter Bootstrap v3*
- JavaScript Library: *jQuery*
- Programming Language: *Python*
- Python Web framework: *Django*
- Python & Django Libraries: *haystack, Pillow*
- APIs: *YouTube Live Streaming API V3, Google Maps API*

6.2.5 Prototype Setup

To install and use thydash event management module:

- The client should use a JavaScript enabled web browser.
- The Django Application should be hosted on python 3 and Django 1.8.7 installed machine having WSGI (web server gateway interface) web server.

6.3 Live Streaming Module

6.3.1 Features

The features that will be handled by live streaming module are:

- Login with Google® account.
- Show created events list.
- Encode live videos.
- Transmit videos to streaming server.

6.3.2 Design Diagram

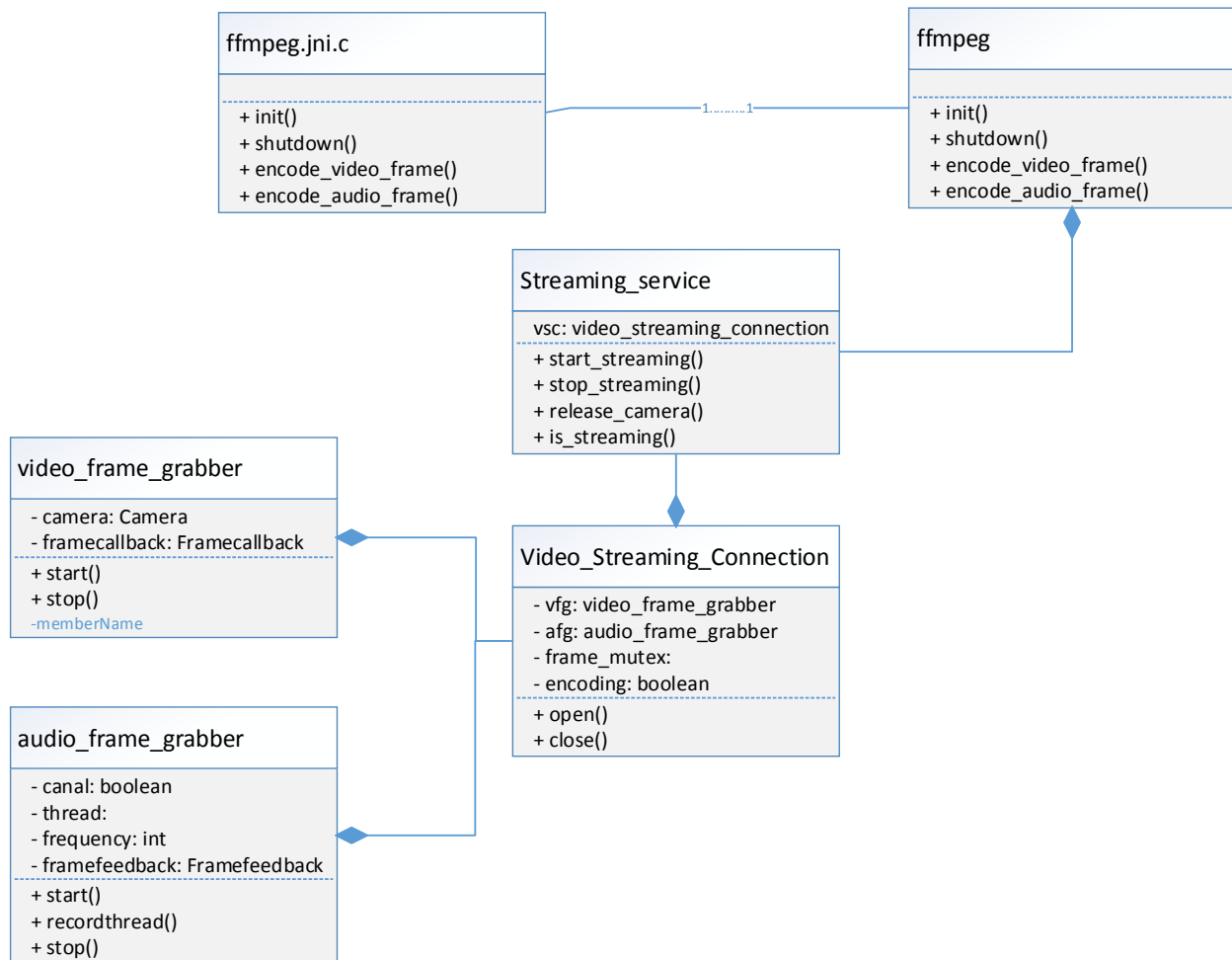


Figure 37 MTV model for Streaming Module

6.3.3 Screenshots

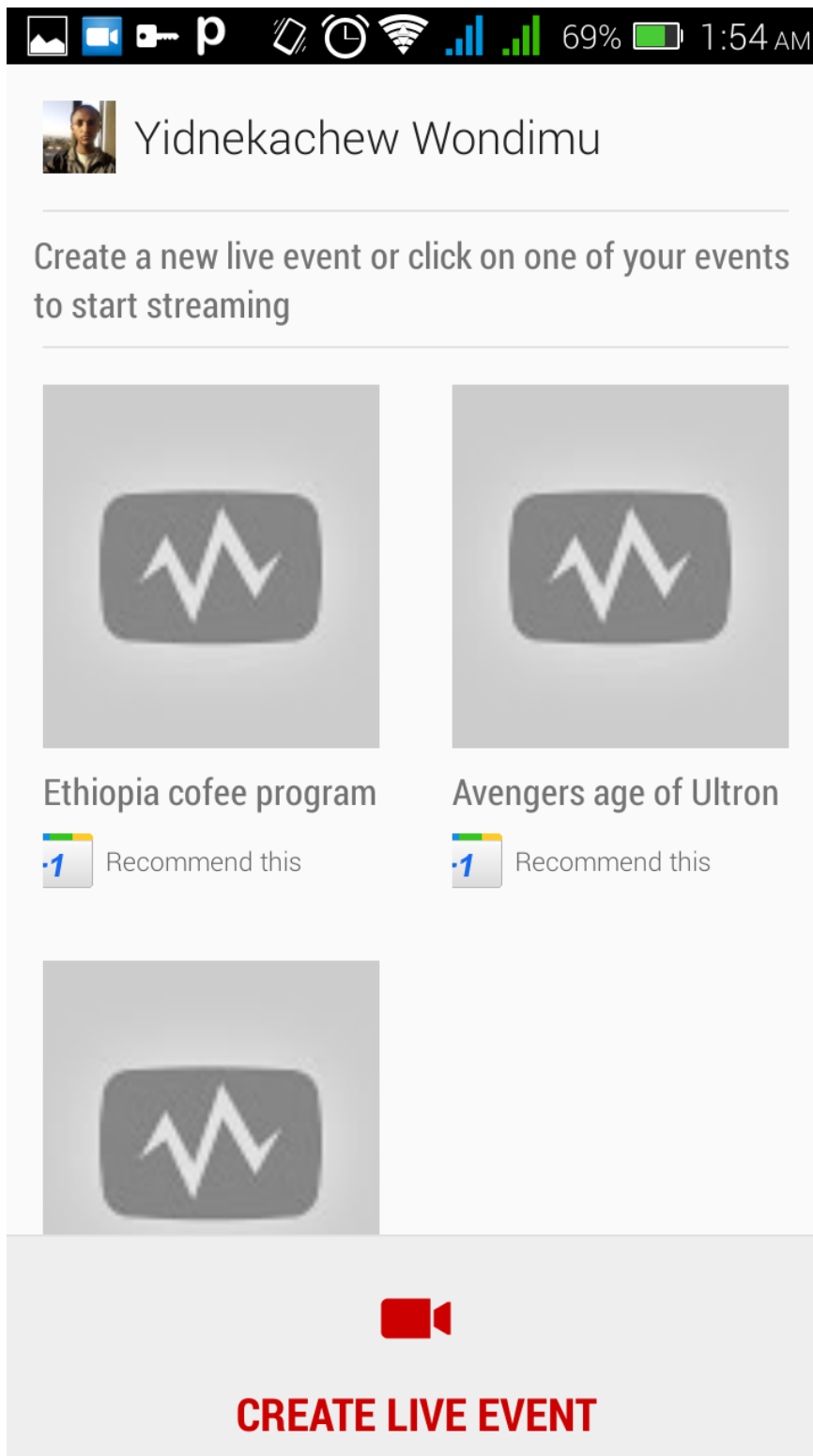


Figure 38: Screenshot Live Events List

6.3.4 Tools and Technologies Used

The tools and technologies that we used to implement user management module are:

Tools

- Android Studio, Android Phone

Technologies

- Encoding Library: *Ffmpeg*
- Shared and Static Libraries Compilation: *GCC*
- Android: *Native Development Kit (NDK)*, *Java Native Interface(JNI)*
- Programming Language: *Java, C*
- Android Libraries: *Volley*, *play-services-plus*, *google-http-client*, *google-api-client-gson*, *play-services-auth*
- APIs: *YouTube Live Streaming API V3*, *Google Plus API*

6.3.5 Prototype Setup

To install and use thydash live streaming module:

- The android device should have ARM architecture.
- The android version should be greater than Android 4.1 (JellyBean)

6.4 Instant Messaging Module

6.4.1 Features

The features that will be handled by instant messaging module are:

- Handle live discussion.

6.4.2 Design Diagram

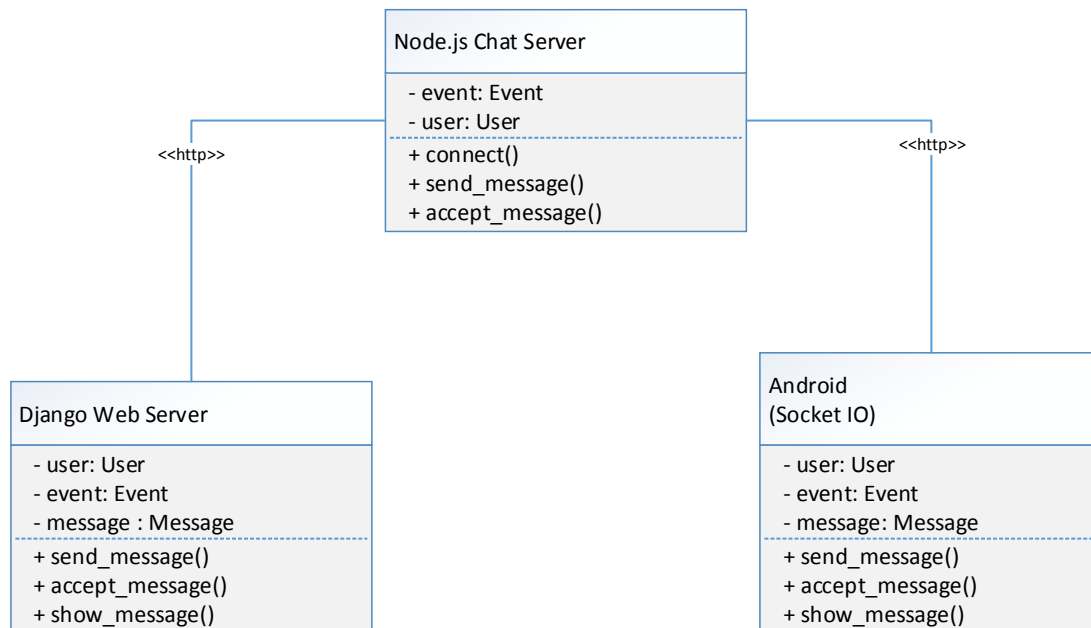


Figure 39 MTV Module for Instant Message module

6.4.3 Tools and Technologies Used

The tools and technologies that we used to implement user management module are:

Tools

- Sublime Text, PyCharm, Android Studio, Android

Technologies

- JavaScript Framework: *NodeJS*
- NodeJS Libraries: *Socket.IO*
- Python Web framework: *Django*
- Programming Language: *JavaScript, Java, Python*
- Android Libraries: *socket.io-client*
- Front end framework: *Twitter Bootstrap v3*
- JavaScript Library: *JQuery*

6.4.4 Prototype Setup

To install and use thydash live streaming module:

- The client should use a JavaScript enabled web browser.
- A chat server which have NodeJS installed.
- The Django Application should be hosted on python 3 and Django 1.8.7 installed machine having WSGI (web server gateway interface) web server.
- The android device should have ARM architecture.
- The android version should be greater than Android 4.1 (JellyBean)

Chapter 7: Testing

“Software never was perfect and won’t get perfect.

But is that a license to create garbage?

The missing ingredient is our reluctance to quantify quality.”

– Boris Beizer²⁵

7.1 Overview

Thydash is a powerful platform providing the intended users (e-Tourist & e-Tour guide) with a capability to travel around the world and get experience with ability to view relevant information such as personal and event details.

The main purpose of this document is to keep track the necessary information required to effectively define the approach to be used in the testing of the project’s product. The test plan document is created during the planning phase of the project. Its intended audience is the project team. Some portions of this document may on occasion be shared with the advisor and other stakeholders whose input/approval into the testing process is needed.

7.2 Objective

The major objectives of testing thydash are; finding defects which may get created by the programmer while developing the software, gaining confidence in and providing information about the level of quality, to prevent defects, to make sure that the end result meets the business and user requirements, to ensure that it satisfies the Business Requirement Specification and System Requirement Specifications, to gain the confidence of the customers by providing them a quality product.

7.3 Scope

The tests referenced herein are written to validate use cases, requirements (both functional and non-functional), system architecture, and object design. The structured tests for object design will be run first as the components of the system are developed. The structured tests to validate the system architecture will be run next as the system is integrated in bottom-up fashion during integration test.

We have conducted various types of testing stages. First while we were developing a feature there was a **unit testing** to check if we were getting the expected result from the module. After that, there was **integration testing** phase in which we made sure whether the different modules are working together flawlessly as they were after unit testing phase. And finally, after finishing

²⁵ **Boris Beizer** is an American software engineer and author. He received his B.S. degree in physics from the City College of New York in 1956, an MS in Electrical Engineering and a PhD in computer science from the University of Pennsylvania in 1966.

the functionalities of the system, we have conducted **system testing** to make sure the whole system is working as per the software requirement specification document.

7.4 Resource (Relation with other documents)

The test document has a relationship to the other documents produced during the development effort of thydash, such as the RAD, SDD, and ODD (Object Design Document).

- ✓ Black box tests relating to use cases are developed from the use case diagram(s) in the RAD (requirements analysis document).
- ✓ Black box tests derived from functional requirements are developed from the requirements lists in the RAD.
- ✓ Performance tests derived from nonfunctional requirements are developed from the nonfunctional requirements in the RAD.
- ✓ Structured (unit/white box) tests are generated from the OOD (Object Design Document). The specific tests are developed from the OOD component diagram of each of the components.
- ✓ Integration tests are developed from the SDD (System/Architecture Design Document). The integration tests generally come from the overall package diagram describing the architecture of the system. The architecture is also used to help in determining the integration test approach. The test environment (hardware/software) is also derived from the SDD.

A visualization of the relationships to the other documents can be seen in the diagram below.

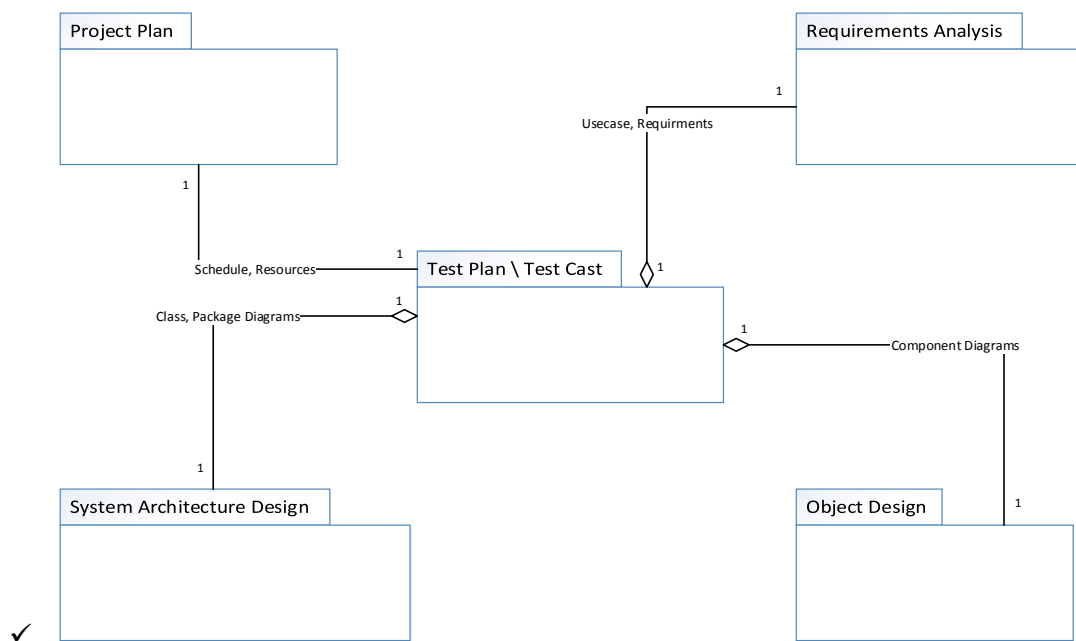


Figure 40 Model of Document Relationship

7.5 Schedule

Unit Testing(UT)

Table 25 Unit Testing 001: Q&A module on web

Unit Testing 001	
Module to be tested	Question and answering module on Web
Features to be tested	<ul style="list-style-type: none"> ▪ Receiving user information from the database (passed) ▪ Displaying user information on question/answer box (passed) ▪ Sending the question/answer(passed) ▪ Save the question/answer to the database (passed) ▪ Whether the question/answer module is working real time or not (passed)
Tested by	Yohannes Aregay

Table 26 Unit Testing 002: Q&A module on android app

Unit Testing 002	
Module to be tested	Question and answering module on Android app
Features to be tested	<ul style="list-style-type: none"> ▪ Receiving user information from the database (passed) ▪ Displaying user information on question/answer box (passed) ▪ Sending the question/answer(passed) ▪ Save the question/answer to the database (passed) ▪ Whether the question/answer module is working real time or not (passed)
Tested by	Yohannes Aregay, Yidnekachew Wondimu

Table 27 Unit Testing 003: Sign Up with Google

Unit Testing 003	
Module to be tested	Sign In / Up with Google

Features to be tested	<ul style="list-style-type: none"> ▪ If the user is already registered, log him/her in. (passed) ▪ If the user isn’t registered, register and log him/her in.(passed)
Tested by	Yidnekachew Wondimu

Table 28 Unit Testing 004: Create Event module on web and YouTube

Unit Testing 004	
Module to be tested	Event Creation module on web and YouTube
Features to be tested	<ul style="list-style-type: none"> ▪ Receive user information from the database (passed) ▪ Save event information in the database (passed) ▪ Sending event information to YouTube channel (passed) ▪ Creates an event in the YouTube channel of the logged in user. (passed)
Tested by	Yidnekachew Wondimu, Melaku Minas

Table 29 Unit Testing 005: Searching Event

Unit Testing 005	
Module to be tested	Searching Events
Features to be tested	<ul style="list-style-type: none"> ▪ Read search query from the form (passed) ▪ Return search results. (passed)
Tested by	Haben Birhane

Table 30 Unit Testing 006: Comment Feature on Past Event

Unit Testing 006	
Module to be tested	Comment feature on past event
Features to be tested	<ul style="list-style-type: none"> ▪ Receive user information from database (passed) ▪ Receive event information from database (passed) ▪ Save comment by a user (passed)

	<ul style="list-style-type: none"> Display previous comments (passed)
Tested by	Haben Birhane

Table 31 Unit Testing 006: Streaming from Android camera

Unit Testing 006	
Module to be tested	Streaming from Android camera
Features to be tested	<ul style="list-style-type: none"> Receive user information in the android app (passed) Record video using the app (passed) Encode the video (passed) Stream live video from android phone (passed) Receive real time video in the web (passed)
Tested by	Yidnekachew Wondimu, Yeabsera Molla

Table 32 Unit Testing 007: Braintree Payment module

Unit Testing 007	
Module to be tested	Braintree Payment module
Features to be tested	<ul style="list-style-type: none"> Perform payment using Braintree payment getaway (passed)
Tested by	<ul style="list-style-type: none"> Yohannes Aregay, Yidnekachew Wondimu

Integration Testing (IT)

Table 33 Integration Testing 001

Integration Testing 001	
Module to be tested	Question and answering module on Android app Question and answering module on Web
Features to be tested	<ul style="list-style-type: none"> Whether a question/answer from Android user is sent to the web users (passed) Whether a question/answer from the Web is sent to the Android user (passed)

Tested by	Yohannes Aregay, Yidnekachew Wondimu
-----------	--------------------------------------

Table 34 Integration Testing 002

Integration Testing 002	
Module to be tested	Event module Braintree payment getaway module
Features to be tested	<ul style="list-style-type: none"> ▪ If a user watches an event if and only if he/she has paid for the event (passed)
Tested by	Haben Birhane, Yohannes Aregay, Yidnekachew Wondimu

Table 35 Integration Testing 003

Integration Testing 003	
Module to be tested	Live streaming module Question and answer module in the android device
Features to be tested	<ul style="list-style-type: none"> ▪ Both the streaming and the question and answer modules are active at the same screen (passed) ▪ Both the streaming and question & answer start and end at the same time. (passed) ▪ Both need to work simultaneously without crushing
Tested by	Haben Birhane, Yohannes Aregay, Yidnekachew Wondimu

Table 36 Integration Testing 004

Integration Testing 004	
Module to be tested	Event module Live streaming module
Features to be tested	<ul style="list-style-type: none"> ▪ Receives event information (passed) ▪ Returns video to the event (passed)
Tested by	Yidnekachew Wondimu, Yeabsera Molla

Table 37 Integration Testing 005

Integration Testing 005	
Module to be tested	User module Live Streaming
Features to be tested	<ul style="list-style-type: none"> Receives user information (passed) Forward Real time video to users (passed)
Tested by	Yidnekachew Wondimu, Yeabsera Molla

7.6 Features to Be Tested or Not To Be Tested

On this section of the test plan document, we will identify all features and combinations of features to be tested. We will also describe all those features that are not to be tested and the reasons for not testing them. Based on this we have identified features to be tested or not to be tested as follows.

A software product should be tested in various ways. But some features or functionalities may not need testing. For example, in our case, we use sign up with Google and pay through Braintree features. As a result, we couldn’t test much on that except whether it is working or not. Based on this we have identified features to be tested or not as follows.

7.6.1 Features to Be Tested

The features and combination of features to be tested are components developed by our project team. Components developed by outsource vendors will not be tested as components. Rather the tests by the vendors will be reviewed by the team, and if they pass review, the component will be tested in-house starting with integration test.

- Create thydash account with Gmail (Sign up with Gmail)
- Logout
- Update Profile
- Event creation with web
- Event streaming from Android phone
- Question and Answering section on web
- Question and Answering section on Android phone
- Commenting section on Web
- Payment system with Braintree Sandbox account

7.6.2 Feature Not To Be Tested

In this test document we will not test third party and off the shelf components, because the components were evaluated and the pros and cons weighed before choosing that component with our software. And infrastructure components such as actual database software utilized is

assumed to work as designed and will not be directly tested for functionality. Therefore, the following features are not going to be tested.

- User authentication, since it is handled by Gmail itself
- Payment on a real account

7.7 Pass/Fail criteria

7.7.1 Component Pass/Fail criteria

Tests executed on components only pass when they satisfy the signatures, constraints, and interfaces dictated by the Object Design Specification for that component. This includes positive tests, negative and stress tests, and boundary tests.

7.7.2 Integration Pass/Fail criteria

Tests executed on integrated components only pass when they satisfy the signatures, constraints, and interfaces dictated by both the object design specification and the system architecture specification. This includes positive tests, negative and stress tests, boundary conditions, and tests that explicitly manipulate the interface environment (such as the physical connection to the database server).

7.7.3 System Pass/Fail criteria

Tests executed against the system use the functional requirements, non-functional requirements, and use cases to determine pass or fail.

7.8 Approach

Unit testing and component testing will be performed on the components as they are developed. Test will be executed using test code in the form of either custom test tools or as an automated suite of tests run against the components

Integrations tests will be performed by both the component testers (developer of the component) as well as the system testers. However, as the integration begins to include GUI level functionality, the tests being run will utilize significantly more manual testing and less automated testing.

System test will require a new set of tools that can measure NFRS²⁶ compliance, such as Selenium testing tool

²⁶ NFRS stands for Non Functional Requirements

7.8.1 Test case alignment with Test phases

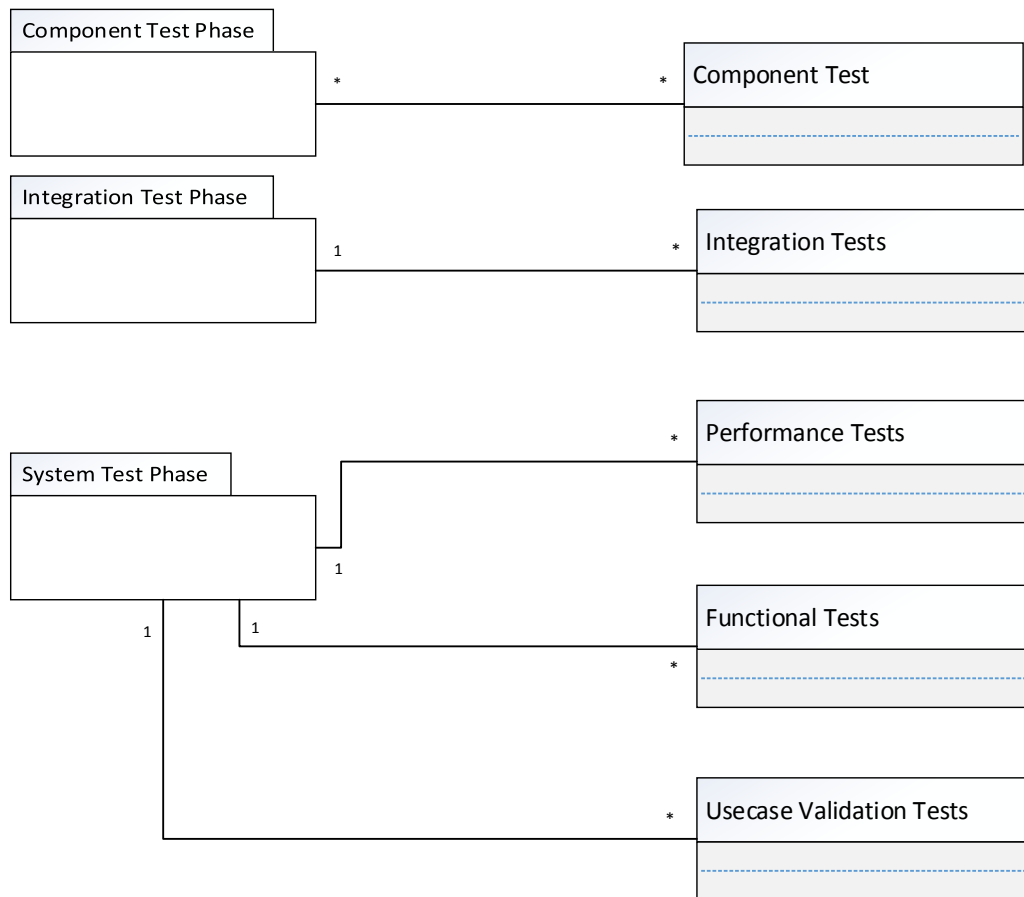


Figure 41 Test case alignment with Test phases

7.9 Test case specification

Table 38 TC001: User Profile - Phone Number Validity

Test Case Name	User profile – Phone Number Validity
Test Case Id	TC 001
Description	Set phone number of a user profile using invalid characters.
Prerequisites	User should be logged in.
Environment Information	Internet connection working.
Steps	1. Open user profile page.

	<ol style="list-style-type: none"> 2. Insert an invalid phone number 3. Click save profile.
Test Data	05xrt913000
Expected Result	Invalid phone number error message
Actual Result	It shows error message.
Remark	PASS

Table 39 TC002: User Profile- Phone Number validity

Test Case Name	User profile – Phone Number Validity
Test Case Id	TC 002
Description	Set phone number of a user profile using valid characters.
Prerequisites	User should be logged in.
Environment Information	Internet connection working.
Steps	<ol style="list-style-type: none"> 1. Open user profile page. 2. Insert an valid phone number 3. Click save profile.
Test Data	+2519111111
Expected Result	Save to database.
Actual Result	It saves the phone no in the profile.
Remark	PASS

Table 40 TC003: Event Creation - Price Validity 0

Test Case Name	Event Creation – Price Validity
Test Case Id	TC 003
Description	Set price of an event to be of negative value.

Prerequisites	User should be logged in. User's YouTube account should be verified and live streaming should be enabled.
Environment Information	Internet connection working.
Steps	<ol style="list-style-type: none"> 1. Open create event page. 2. Insert a negative price. 3. Click create event.
Test Data	-1 USD
Expected Result	Show error message.
Actual Result	It doesn't accept negative value for price of event.
Remark	PASS

Table 41 TC004: Event Creation - Price Validity 1

Test Case Name	Event Creation – Price Validity
Test Case Id	TC 004
Description	Set price of an event to be zero USD.
Prerequisites	User should be logged in. User's YouTube account should be verified and live streaming should be enabled.
Environment Information	Internet connection working.
Steps	<ol style="list-style-type: none"> 1. Open create event page. 2. Insert zero USD 3. Click create event.
Test Data	0 USD
Expected Result	Create free event.

Actual Result	It has created free event
Remark	PASS

Table 42 TC005: TC005: Event Creation - Price Validity 2

Test Case Name	Event Creation – Price Validity
Test Case Id	TC 005
Description	Set price of an event to be positive USD.
Prerequisites	User should be logged in. User's YouTube account should be verified and live streaming should be enabled.
Environment Information	Internet connection working.
Steps	<ol style="list-style-type: none"> 1. Open create event page. 2. Insert positive USD 3. Click create event.
Test Data	1 USD
Expected Result	Create paid event.
Actual Result	It creates paid event
Remark	PASS

Table 43 TC006: Event Creation - Event Date Validity 0

Test Case Name	Event Creation – Event Date Validity
Test Case Id	TC 006
Description	Set event date to be before today.
Prerequisites	User should be logged in. User's YouTube account should be verified and live streaming should

	be enabled.
Environment Information	Internet connection working.
Steps	<ol style="list-style-type: none"> 1. Open create event page. 2. Select event date to be before today. 3. Click create event.
Test Data	May 30, 2016
Expected Result	Create event for today.
Actual Result	User can only access date starting from today and onwards.
Remark	PASS

Table 44 TC007: Event Creation – Event Date Validity 1

Test Case Name	Event Creation – Event Date Validity
Test Case Id	TC 007
Description	Set event date to be for today.
Prerequisites	<p>User should be logged in.</p> <p>User's YouTube account should be verified and live streaming should be enabled.</p>
Environment Information	Internet connection working.
Steps	<ol style="list-style-type: none"> 4. Open create event page. 5. Select event date to be for today. 6. Click create event.
Test Data	May 31, 2016
Expected Result	Create event for today.
Actual Result	Created an event for today.
Remark	PASS

Table 45 TC008: Event Creation - Event Date Validity 2

Test Case Name	Event Creation – Event Date Validity
Test Case Id	TC 008
Description	Set event date to be after today.
Prerequisites	User should be logged in. User's YouTube account should be verified and live streaming should be enabled.
Environment Information	Internet connection working.
Steps	<ol style="list-style-type: none"> 1. Open create event page. 2. Select event date to be after today. 3. Click create event.
Test Data	July 05, 2016
Expected Result	Create event for July 5, 2016.
Actual Result	Creates only for 2 months forward.
Remark	PASS

Table 46: TC009: Event Creation - Time Range Validity 0

Test Case Name	Event Creation – Time Range Validity
Test Case Id	TC 009
Description	Set event start time to be after end time.
Prerequisites	User should be logged in. User's YouTube account should be verified and live streaming should be enabled.
Environment Information	Internet connection working.

Steps	<ol style="list-style-type: none"> 1. Open create event page. 2. Make start time to be after end time. 3. Click create event.
Test Data	{starting time: 09:00, ending time: 08:00}
Expected Result	Show error message.
Actual Result	It doesn't show error message.
Remark	PASS

Table 47 TC010: Event Creation - Time Range Validity 1

Test Case Name	Event Creation – Time Range Validity
Test Case Id	TC 010
Description	Set event start time to be before end time.
Prerequisites	<p>User should be logged in.</p> <p>User's YouTube account should be verified and live streaming should be enabled.</p>
Environment Information	Internet connection working.
Steps	<ol style="list-style-type: none"> 1. Open create event page. 2. Make start time to be before end time. 3. Click create event.
Test Data	{starting time: 08:00, ending time: 09:00}
Expected Result	Event Successfully created.
Actual Result	Created event successfully
Remark	PASS

Table 48 TC011: Event Creation - Time Range Validity 2

Test Case Name	Event Creation – Time Range Validity
----------------	---

Test Case Id	TC 011
Description	Set event start time to be the same as end time.
Prerequisites	User should be logged in. User's YouTube account should be verified and live streaming should be enabled.
Environment Information	Internet connection working.
Steps	<ol style="list-style-type: none"> 1. Open create event page. 2. Make start time to be the same as end time. 3. Click create event.
Test Data	{starting time: 18:00, ending time: 18:00}
Expected Result	Show error message.
Actual Result	It has created an event.
Remark	PASS

Table 49 TC012: Event Creation - Location Validity 0

Test Case Name	Event Creation – Location Validity
Test Case Id	TC 012
Description	Try to create an event without inserting location.
Prerequisites	User should be logged in. User's YouTube account should be verified and live streaming should be enabled.
Environment Information	Internet connection working. Google Map should be shown in the page.
Steps	<ol style="list-style-type: none"> 1. Open create event page. 2. Make location of the event to be empty.

	3. Click create event.
Test Data	{location: empty}
Expected Result	Show error message.
Actual Result	Show error message. And notify user to fill location
Remark	PASS

Table 50 TC013: Event Creation - Location Validity 1

Test Case Name	Event Creation – Location Validity
Test Case Id	TC 013
Description	Try to create an event by selecting a location.
Prerequisites	User should be logged in. User's YouTube account should be verified and live streaming should be enabled.
Environment Information	Internet connection working. Google Map should be shown in the page.
Steps	<ol style="list-style-type: none"> 1. Open create event page. 2. Make location of the event to be empty. 3. Click create event.
Test Data	{location : Addis Ababa Ethiopia, latitude: 8.37, longitude: 38.51}
Expected Result	Successfully create an event.
Actual Result	It has successfully created an event
Remark	PASS

Table 51 TC014: Joining Event - Handling Payment 0

Test Case Name	Joining Event – Handling Payment
Test Case Id	TC 014

Description	Try to join a paid event.
Prerequisites	User should be logged in. The event should be created by another user. The event should be an upcoming event.
Environment Information	Internet connection working.
Steps	<ol style="list-style-type: none"> 1. Open event details page. 2. Click Join event button. 3. Fill out working payment information to Braintree's payment handling page. 4. Click pay button.
Test Data	
Expected Result	Transfer the event price from the joining person's account to thydash account and successfully join the event.
Actual Result	Successfully transferred account and successfully joined an event.
Remark	PASS

Table 52 TC015: Deleting Paid Event

Test Case Name	Deleting Paid Event
Test Case Id	TC 015
Description	Try to delete a paid event.
Prerequisites	User should be logged in. The event should be created by logged in user. The event should be an upcoming event.
Environment Information	Internet connection working.
Steps	<ol style="list-style-type: none"> 1. Open event details page.

	2. Click delete event button.
Test Data	
Expected Result	Pay back the payments to the joined users and delete the event.
Actual Result	Deletes the event without paying back.
Remark	FAIL

Table 53 TC016: Deleting Free Event

Test Case Name	Deleting Free Event
Test Case Id	TC 016
Description	Try to delete a free event.
Prerequisites	User should be logged in. The event should be created by logged in user. The event should be an upcoming event.
Environment Information	Internet connection working.
Steps	1. Open event details page. 2. Click delete event button.
Test Data	
Expected Result	Delete the event and notify the joined people that the event has been deleted.
Actual Result	Deletes the event & notifies the joined people
Remark	PASS

Table 54 TC017: Event Streaming from Android phone - Account Validity 0

Test Case Name	Event Streaming from Android phone - Account validity
Test Case Id	TC 017
Description	Log in with unregistered Gmail account
Prerequisites	
Environment Information	Internet connection should be available Google Play Service must up to date Android version should be >= Android 4.1 (JellyBean)
Steps	1. open the thydash app in the android phone 2. choose an email account which is not registered in thydash 3. click sign in button
Test Data	tdksgm@gmail.com (unregistered in thydash)
Expected Result	The system should register the user in thydash account
Actual Result	It registers to thydash.
Remark	PASS

Table 55 TC018: Event Streaming from Android phone - Account Validity 1

Test Case Name	Event Streaming from Android phone - Account validity
Test Case Id	TC 018
Description	User registered with more than one Gmail account in play store
Prerequisites	
Environment Information	Internet connection should be available Google Play Service must up to date Android version should be >= Android 4.1 (JellyBean)

Steps	1. open the thydash app in the android phone 2. Account chooser dialog box will pop up. 3. sign in with one of the registered Gmail accounts 4. click sign in button
Test Data	Hab177@gmail.com
Expected Result	The user logs in to the thydash app
Actual Result	It logs in
Remark	PASS

Table 56 TC019: Event Streaming from Android phone - Account validity

Test Case Name	Event Streaming from Android phone - Account validity
Test Case Id	TC 019
Description	User registered with one Gmail account in play store
Prerequisites	
Environment Information	Internet connection should be available Google Play Service must up to date Android version should be \geq Android 4.1 (JellyBean)
Steps	1. open the thydash app in the android phone 2. account chooser dialog box with one account will pop up 2. sign in with the registered Gmail accounts 3. click sign in button
Test Data	Hab177@gmail.com
Expected Result	The user logs in to the thydash app
Actual Result	Yes it registers
Remark	PASS

Table 57 TC020: Event Streaming from Android phone - Retrieving Event list

Test Case Name	Event Streaming from Android phone - Retrieving Event list
Test Case Id	TC 020
Description	After logging in to the app, User haven't created any event yet.
Prerequisites	User should be logged in.
Environment Information	Internet connection should be available Google Play Service must up to date Android version should be >= Android 4.1 (JellyBean)
Steps	1. open the thydash app in the android phone 2. account chooser dialog box with one account will pop up 3. choose an account and sign in with it 4. Empty list and Create Event button will be displayed
Test Data	Hab177@gmail.com
Expected Result	The event list screen will be displayed
Actual Result	It displays event list
Remark	PASS

Table 58 TC021: Event Streaming from Android phone - Retrieving Event list

Test Case Name	Event Streaming from Android phone - Retrieving Event list
Test Case Id	TC 021
Description	After logging in to the app, User have created events in the web
Prerequisites	User should be logged in.
Environment Information	Internet connection should be available Google Play Service must up to date Android version should be >= Android 4.1 (JellyBean)

Steps	<ol style="list-style-type: none"> 1. open the thydash app in the android phone 2. account chooser dialog box with one account will pop up 3. choose an account and sign in with it 4. list of events and Create Event button will be displayed
Test Data	Hab177@gmail.com
Expected Result	The event list screen will be displayed
Actual Result	It shows list of events which are created through web also.
Remark	PASS

Table 59 TC022: Event Streaming from Android phone - Streaming an Event

Test Case Name	Event Streaming from Android phone - Streaming an Event
Test Case Id	TC 022
Description	After logging in to the app, User have an event to broadcast
Prerequisites	User should be logged in.
Environment Information	<p>Internet connection should be available</p> <p>Google Play Service must up to date</p> <p>RTMP protocol should be working on the network</p> <p>Android version should be >= Android 4.1 (JellyBean)</p>
Steps	<ol style="list-style-type: none"> 1. open the thydash app in the android phone 2. account chooser dialog box with one account will pop up 3. choose an account and sign in with it 4. list of events and Create Event button will be displayed 5. select an event to stream 6. Click in the Start button.
Test Data	- Ethiopian Coffee ceremony event started streaming
Expected Result	The event starts streaming to the YouTube server.

Actual Result	It starts streaming selected event
Remark	PASS

Table 60 TC023: Q&A section on the web - Start Instant Messaging

Test Case Name	Question & Answering section on the web - Start Instant Messaging
Test Case Id	TC 023
Description	After logging in on the web, user clicked on the watch live button and live streaming started
Prerequisites	User should be logged in.
Environment Information	Chat server should be running
Steps	<ol style="list-style-type: none"> 1. User signed in to the thydash web using Gmail account 2. Click on an event to watch live 3. live streaming started 4. User starts chatting(Q&A)
Test Data	- Ethiopian Coffee ceremony event started streaming
Expected Result	The chat(Q&A) section is active, User starts instant messaging
Actual Result	Can make discussions when the event is live.
Remark	PASS

Table 61 TC024: Q&A section on the web Live Streaming over

Test Case Name	Question & Answering section on the web - Live streaming over
Test Case Id	TC 024
Description	After logging in on the web, user watched the event live and the event is over.
Prerequisites	User should be logged in.

Environment Information	Chat server should be running
Steps	<ol style="list-style-type: none"> 1. User signed in to the thydash web using Gmail account 2. Click on an event to watch live 3. live streaming started 4. live streaming is over
Test Data	- Ethiopian Coffee ceremony event finished streaming
Expected Result	The chat(Q&A) section will also be over, and it will be Inactive.
Actual Result	Discussion waits until the time set for the event is over, not the live streaming.
Remark	PASS

Table 62 TC025: Q&A section on the Android - Start Instant Messaging

Test Case Name	Question & Answering section on the Android- Start Instant Messaging
Test Case Id	TC 025
Description	After logging in on the Android device, user clicked on the Start live button and live streaming started
Prerequisites	User should be logged in.
Environment Information	Chat server should be running Google Play Service must up to date RTMP protocol should be working on the network Android version should be >= Android 4.1 (JellyBean)
Steps	<ol style="list-style-type: none"> 1. User signed in to the thydash web using Gmail account 2. Click on an event to start live 3. live streaming started 4. e-Tour starts chatting(Q&A)

Test Data	- Ethiopian Coffee ceremony event started streaming
Expected Result	The chat(Q&A) section is active, e-Tour starts to receive instant messaging and send messages.
Actual Result	It works as required.
Remark	PASS

Table 63 TC026: Q&A section on the Android - Live Streaming over

Test Case Name	Question & Answering section on the Android - Live streaming over
Test Case Id	TC 026
Description	After logging in on the Android device, user streamed the event live and the event is over.
Prerequisites	User should be logged in.
Environment Information	Chat server should be running Google Play Service must up to date RTMP protocol should be working on the network Android version should be >= Android 4.1 (JellyBean)
Steps	1. User signed in to the thydash web using Gmail account 2. Click on an event to Start live 3. live streaming started 4. live streaming is over
Test Data	- Ethiopian Coffee ceremony event finished streaming
Expected Result	The chat(Q&A) section will also be over, and it will be Inactive.
Actual Result	It stops discussion when the streaming is over.
Remark	PASS

7.10 Estimated risk and contingency plan

Table 64 Estimated risk and contingency plan

Risk	Probability	Risk Type	Owner	Contingency/Mitigation Approach
Unable to acquire the necessary skilled personnel as the components become ready to test.	15%	Personnel Schedule	Team Leader	Resources for components will be split between the existing resources. Schedule must be adjusted accordingly.
Unable to acquire some of the necessary hardware and software required for integration and system testing	20%	Equipment	Team	Utilize the existing acquire Hardware, and look out for trial version of the software required, or use cracked.
Third party services utilized in the system become unavailable during testing	60%	Third Party	School	Test it somewhere else.
Turnover	5%	Personnel	Team Leader	If a single member of team decides to leave, the team leader will reshuffle and reassign team members to components to be tested

7.11 Conclusion

Software testing cannot show the absence of errors and defects, it can show only that software errors and defects are present. So in this phase of the project we were able to identify problems (bugs) in our product and by identifying them we could make our product much better. We were able to verify whether we are following as per the SRS and BRS.

Conclusion and Recommendation

Tourism, as an industry, is a big source of income for many countries. And for people who tour, it is a leisure activity. But, in this era of information technology, it's hard to know that people are bound from visiting because of their time shortage, income level and disabilities. You know, it makes no sense to have only tens or hundreds of people visit our moon. We all want to get an understanding of what exists there.

The idea behind developing thydash is solving this problem by providing an alternative to the traditional tourism. We wanted to provide a platform which enables people to visit somewhere else from their location. For this to happen, eTourists join events created by eTour-guides and watch the live streaming session and make discussions in real time.

We, now, have that solution we have proposed for. We can now start live streaming at a click of a button. We can, at the same time, discuss with the eTour guide in real-time. Only some big names can be mentioned that have achieved this as of now; Twitter's Periscope and Meerkat. Though we have achieved it, there were many hardships that we have come through.

The major problem we have faced among these was an internet connection problem. Since our project is network intensive, we required the internet to work flawlessly and UDP specifically RTMP protocols to be allowed. It was blocking to have any of these problems, which we were facing for most of the time in the semester. We have solved this problem by going to Addis, internet cafes and hotels for development.

The other problem we've faced was lack of sufficient information in the technologies, APIs and libraries we were using (Django, NodeJS, and Ffmpeg). Since most of the technologies we used are new, getting enough information about them was not as easy task. But by going through their documentations, we were able to reach where we are now.

We've mentioned the bright sides of thydash until now. But, we are not satisfied with what we have achieved so far. We think thydash can get a lot better with some enhancements such as using our own streaming server, making it to work with less internet bandwidth, developing the application for iOS smart phones, and making it commercially feasible.

Finally, we would like to give some recommendations about the overall process of the project. Currently, most of the software firms are going the agile way. There is no better place to familiarize candidate graduates to this new and better way of developing software products. So, the software development life cycle that is allowed for senior project shouldn't be bound to the waterfall model. The other thing to mention is that the process of keep tracking senior projects lacks detailed analysis of what the project requires.

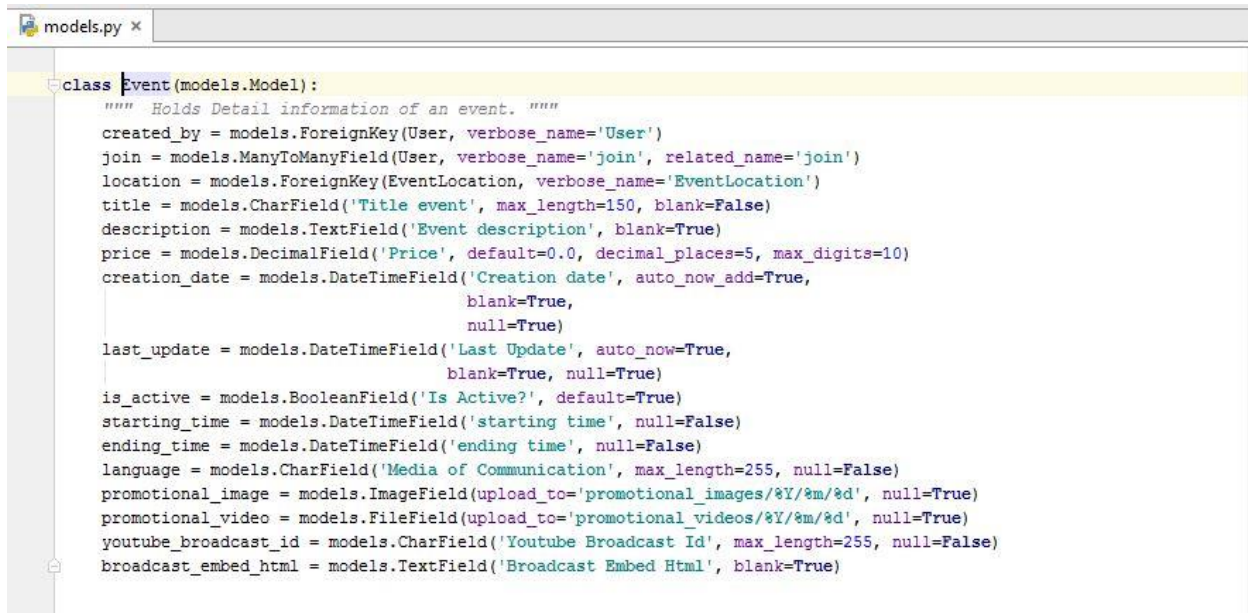
Of course, the idea of thydash is not limited to this senior project. We would like to make it a usable product. Anyone interested on contributing to this idea can contact us through our email (thydash.developers@gmail.com).

References

- Greenberg, A. D., & Zanetis, J. (2012). *The Impact of Broadcast and Streaming Video*. Cisco Systems Inc. to Wainhouse Research.
- Huifang Sun, A. V. (2007, February). *An Overview of Scalable Video Streaming*.
- Bailey, J. M. (2011). *Live Video Streaming from Android - Enabled Devices to Web Browsers*. Florida: University of South Florida.
- Hu, P., Nick, S., Jadwiga, I., & Marius, P. (n.d.). *Share your view: wireless multi-hop video streaming using Android phones*. Queensland, Australia: The University of Queensland.
- Pires, K. (2015). *Delivery and transcoding for large scale live streaming systems*. Paris: Universite Pierre et Marie Curie.
- L. B., B. D., & J. C. (2002). *Evaluating the use of the Web for tourism marketing: a case study from New Zealand*. *Tourism Management*, 557-561.
- Hawkins, D. E., Leventhal, M., & Oden, W. L. (1996). *The Virtual Tourism Environment. Utilisation of Information Technology to Enhance Strategic Travel Marketing*. *PROGRESS IN TOURISM AND HOSPITALITY RESEARCH*, 2, 223-228.
- Buhalis, D., & Deimezi, O. (n.d.). *eTourism Developments in Greece: Information Communication Technologies adoption for*.
- Almer, A., & Stelzel, H. (2008). *Multimedia Visualization of GeoInformation for Tourism Regions based on Remote Sensing Data*.
- Anand Bethapudi (2013). *The Role of ICT in Tourism Industry*, *Journal of Applied Economics and Business*, Vol. 1, Issue 4, pp. 67-79
- Bernd Bruegge, Allen H. Dutoit (2010). *Object Oriented Software Engineering Using UML, Patterns and Java Third Edition*.
- Gary B. Shelly, Harry J. Rosenblatt (2012). *System Analysis and Design Ninth Edition*
- Stephen R. Palmer, John M. Felsing (2002). *A Practical Guide to Feature Driven Development*

Annex

I. Code Snippets



```
models.py ×  
  
class Event(models.Model):  
    """ Holds Detail information of an event. """  
    created_by = models.ForeignKey(User, verbose_name='User')  
    join = models.ManyToManyField(User, verbose_name='join', related_name='join')  
    location = models.ForeignKey(EventLocation, verbose_name='EventLocation')  
    title = models.CharField('Title event', max_length=150, blank=False)  
    description = models.TextField('Event description', blank=True)  
    price = models.DecimalField('Price', default=0.0, decimal_places=5, max_digits=10)  
    creation_date = models.DateTimeField('Creation date', auto_now_add=True,  
                                         blank=True,  
                                         null=True)  
    last_update = models.DateTimeField('Last Update', auto_now=True,  
                                       blank=True, null=True)  
    is_active = models.BooleanField('Is Active?', default=True)  
    starting_time = models.DateTimeField('starting time', null=False)  
    ending_time = models.DateTimeField('ending time', null=False)  
    language = models.CharField('Media of Communication', max_length=255, null=False)  
    promotional_image = models.ImageField(upload_to='promotional_images/%Y/%m/%d', null=True)  
    promotional_video = models.FileField(upload_to='promotional_videos/%Y/%m/%d', null=True)  
    youtube_broadcast_id = models.CharField('Youtube Broadcast Id', max_length=255, null=False)  
    broadcast_embed_html = models.TextField('Broadcast Embed Html', blank=True)
```

Figure 42: Code Snippet - Event Model

```
{% block content %}
<div class="col-md-9">
  <div class="main-grids">

    <div class="top-grids">
      <div class="recommended-info">
        <h3 class="thydash-color">My Events</h3>
      </div>
      {% for event in list_of_my_events %}
        <div class="col-md-4 resent-grid recommended-grid slider-top-grids">
          <div class="resent-grid-img recommended-grid-img">
            <a href="events/details/{{ event.id }}">
              {% if event.promotional_image != None and event.promotional_image != "" %}
                
              {% else %}
                
              {% endif %}
            </a>

            <div class="timebar">
              {% if not event.price > 0 %}
                <span class="pull-left" style="...">Free event</span>
              {% endif %}
              <span class="author author-info pull-right">By: {{ event.created_by.first_name|capfirst }}</span>
            <br/>
            <span class="pull-right"><span class="glyphicon glyphicon-facetime-video"></span>
              <b>{{ event.starting_time|timeuntil }}</b>
            </span>
            <span class="pull-left"><span class="glyphicon glyphicon-time"></span>
              <b>{{ event.starting_time|timesince:event.ending_time }}</b>
            </span>
          </div>
        </div>
      {% endfor %}
    </div>
  </div>
</div>
```

Figure 43: Code Snippet - Create Event Template

```
def insert_broadcast(youtube, options):
    insert_broadcast_response = youtube.liveBroadcasts().insert(
        part="snippet,status,contentDetails",
        body=dict(
            kind="youtube#liveBroadcast",
            snippet=dict(
                title=options['broadcast_title'],
                scheduledStartTime=options['start_time'],
                scheduledEndTime=options['end_time'],
            ),
            status=dict(
                privacyStatus=options['privacy_status']
            ),
            contentDetails=dict(
                monitorStream=dict(
                    enableMonitorStream=False,
                ),
                enableEmbed=True,
                enableArchive=True,
            ),
        ),
    ).execute()
```

Figure 44: Code Snippet - Insert Broadcast Method

```
public class ChatMessage {  
    private long id;  
    private boolean isMe;  
    private String message;  
    private Long userId;  
    private String dateTime;  
  
    public long getId() {  
        return id;  
    }  
  
    public void setId(long id) {  
        this.id = id;  
    }  
  
    public boolean getIsme() {  
        return isMe;  
    }  
  
    public void setMe(boolean isMe) {  
        this.isMe = isMe;  
    }  
  
    public String getMessage() {  
        return message;  
    }  
  
    public void setMessage(String message) {  
        this.message = message;  
    }  
  
    public long getUserId() {  
        return userId;  
    }  
}
```

Figure 45: Code Snippet - Chat Message class


```
@Override
public void open(String url, Camera camera, Surface previewSurface) {
    Log.d(MainActivity.APP_NAME, "open");

    videoFrameGrabber = new VideoFrameGrabber();
    videoFrameGrabber.setFrameCallback(new VideoFrameGrabber.FrameCallback() {
        @Override
        public void handleFrame(byte[] yuv_image) {
            if (encoding) {
                synchronized (frame_mutex) {
                    int encoded_size = Ffmpeg.encodeVideoFrame(yuv_image);
                    Log.d(MainActivity.APP_NAME, "Encoded video! Size = " + encoded_size);
                }
            }
        }
    });

    audioFrameGrabber = new AudioFrameGrabber();
    audioFrameGrabber.setFrameCallback(new AudioFrameGrabber.FrameCallback() {
        @Override
        public void handleFrame(short[] audioData, int length) {
            if (encoding) {
                synchronized (frame_mutex) {
                    int encoded_size = Ffmpeg.encodeAudioFrame(audioData, length);
                    Log.d(MainActivity.APP_NAME, "Encoded audio! Size = " + encoded_size);
                }
            }
        }
    });

    synchronized (frame_mutex) {
        Size previewSize = videoFrameGrabber.start(camera);
        audioFrameGrabber.start(AUDIO_SAMPLE_RATE);

        int width = previewSize.width;
        int height = previewSize.height;
        encoding = Ffmpeg.init(width, height, AUDIO_SAMPLE_RATE, url);

        Log.i(MainActivity.APP_NAME, "Ffmpeg.init() returned " + encoding);
    }
}
```

Figure 46: Code Snippet - VideoStreaming connection

II. Algorithms and Standards

Search: Elastic Search

Elasticsearch is a real-time distributed search and analytics engine. It allows you to explore your data at a speed and at a scale never before possible. It is used for full-text search, structured search, analytics, and all three in combination.

Wikipedia uses Elasticsearch to provide full-text search with highlighted search snippets, and search-as-you-type and did-you-mean suggestions. The Guardian uses it to combine visitor logs with social -network data to provide real-time feedback to its editors about the public's response to new articles. Stack Overflow combines full-text search with geolocation queries and uses more-like-this to find related questions and answers. GitHub uses it to query 130 billion lines of code.

Video Encoding: H.264

H.264 is the most widely used codec on the planet, with significant penetration in optical disc, broadcast, and streaming video markets. However, many uses of H.264 are subject to royalties, something that should be considered prior to its adaption. Other factors to consider include comparative quality against other available technologies, like Google's WebM, as well as the general availability of decoding capabilities on target platforms and devices.

H.264 is a video compression technology, or codec, that was jointly developed by the International Telecommunications Union (as H.264) and International Organization for Standardization/International Electrotechnical Commission Moving Picture Experts Group (as MPEG-4 Part 10, Advanced Video Coding, or AVC). Thus, the terms H.264 and AVC mean the same thing and are interchangeable.

Audio Encoding: AAC

Advanced Audio Coding (AAC) is an audio coding standard for lossy digital audio compression. Designed to be the successor of the MP3 format, AAC generally achieves better sound quality than MP3 at similar bit rates.

AAC has been standardized by ISO and IEC, as part of the MPEG-2 and MPEG-4 specifications. Part of the AAC known as High Efficiency Advanced Audio Coding (HE-AAC) which is part of MPEG-4 Audio is also adopted into digital radio standards like DAB+ and Digital Radio Mondiale, as well as mobile television standards DVB-H and ATSC-M/H.

AAC supports inclusion of 48 full-bandwidth (up to 96 kHz) audio channels in one stream plus 16 low frequency effects (LFE, limited to 120 Hz) channels, up to 16 "coupling" or dialog channels, and up to 16 data streams. The quality for stereo is satisfactory to modest requirements at 96 kbit/s in joint stereo mode; however, hi-fi transparency demands data rates of at least 128 kbit/s (VBR). The MPEG-2 audio tests showed that AAC meets the requirements referred to as "transparent" for the ITU at 128 kbit/s for stereo, and 320 kbit/s for 5.1 audio.

Color Space: YUV

YUV colorspace is a bit unusual. The Y component determines the brightness of the color (referred to as luminance or luma), while the U and V components determine the color itself (the chroma). Y ranges from 0 to 1 (or 0 to 255 in digital formats), while U and V range from -0.5 to 0.5 (or -128 to 127 in signed digital form, or 0 to 255 in unsigned form). Some standards further limit the ranges so the out-of-bounds values indicate special information like synchronization.

One neat aspect of YUV is that you can throw out the U and V components and get a grey-scale image. Since the human eye is more responsive to brightness than it is to color, many lossy image compression formats throw away half or more of the samples in the chroma channels to reduce the amount of data to deal with, without severely destroying the image quality.

Instant Messaging: Socket.IO

Socket.IO is an event-based bi-directional communication layer for real-time web applications, built atop Engine.IO. It abstracts many transports, including AJAX long-polling and WebSockets, into a single API. It allows developers to send and receive data without worrying about cross-browser capability.

It is used by Slack, a workplaces messaging platform, and Yammer, Microsoft's corporate messaging platform for instant messaging purposes.