Table of Contents

[***1.*** ***Overview*** 3](#_Toc501192118)

[***2.*** ***Scope and Objectives*** 4](#_Toc501192119)

[**2.1.** **Objectives** 4](#_Toc501192120)

[**2.2.** **Scope** 4](#_Toc501192121)

[**2.2.1.** **Included** 4](#_Toc501192122)

[**2.2.2.** **Not included** 4](#_Toc501192123)

[***3.*** ***Organization*** 5](#_Toc501192124)

[***4.*** ***Work breakdown structure*** 6](#_Toc501192125)

[**4.1.** **Work Breakdown Structure:** 6](#_Toc501192126)

[**4.2.** **Schedule and Milestones:** 10](#_Toc501192127)

[**4.3.** **Budget** 12](#_Toc501192128)

[**4.4.** **Development Process:** 13](#_Toc501192129)

[**4.5.** **Development Environment:** 14](#_Toc501192130)

[**4.6.** **Measurement Program** 15](#_Toc501192131)

[***5.*** ***Risk Managerment:*** 15](#_Toc501192132)

[**5.1.** **Project risk:** 15](#_Toc501192133)

[**5.2.** **Product risk:** 15](#_Toc501192134)

[**5.3.** **Bussiness risk:** 15](#_Toc501192135)

[***6.*** ***Delivery Plan:*** 16](#_Toc501192136)

[***7.*** ***Abbreviations and Definitions*** 19](#_Toc501192137)

[***8.*** ***References*** 19](#_Toc501192138)

1. ***Overview***

Nowadays, movie and TV series are essential and irreplaceable parts of the people’s everyday entertainment. With the rapid development of the entertainment industry, there are various ways of watching movie and TV series. One of the fastest developing methods is video streaming with plenty of services such as Netflix, Amazon Prime, phimmoi and so on. But the problem is that some charge a very high subscription fee, some don’t but are embedded with annoying ads. With this in mind, we have decided to develop our own movie streaming service which is free, fast and convenient for users.

Our movies watching website have a database of movies information, movies categorized by genre and each of them has a unique name and id to easily access.

The system allows all guests view and search movies by their name, id and genre. If guest wants to have premium features, they could register account.

Each user will have a profile page containing age, avatar, favorite movies, genre and movies they watched. They can edit their information at any time they want. The biggest differences between users and guests are the ability to watch the most updated movies and comments on the movies.

In terms of the administrator, he/ she is able to upload the new movies, delete the obsolete movies, edit movies information, ban the account which has the inappropriate comments

1. ***Scope and Objectives***
   1. **Objectives**

* Fast streaming speed: One of the most important aspect of a movie streaming services is its streaming speed. Thus, we aim to deliver a website with an adequate speed for most users.
* A beautiful and responsive UI: A website makes the best impression if it is a beautiful website and can scale the size of dimension of website based on the device user interacts
* An appropriate database design: website can handle substantial amount of user, their comments and their requests
* Ease of choosing movies: user easily choose their favorite movies
* Watching movies without interruption: user opts for resolution of movie they are watching based on their bandwidth
* Meet the demand for any types of user: if they are the intensive user who demand updated and high-resolution movie, they are required to buy the premium account. Otherwise, website provides free account for the other
  1. **Scope**
     1. **Included**
* The system is designed to run on Apache Tomcat
* The system is just able to allow the certain amount of access simultaneously. It is not designed to support a large number of access; neither is it designed to support a large database.
* The system is able to save the progress on the movie they are watching
* The system will have a comment section on every movie
  + 1. **Not included**
* Live chat
* Recommender System
* Other elements not going to include in the projects

1. ***Organization***

Our team has 4 members, each one has to handle some roles

|  |  |  |
| --- | --- | --- |
| **Name** | **Role** | **Email** |
| Lê Võ Thanh Duy | Databse Designer, Programmer, Report writer |  |
| Phạm Sơn Hải | Programmer, Manager |  |
| Lưu Minh Quân | UI Designer, Programmer, Report writer |  |
| Mai Lâm Trọng Nguyên | Tester, Report writer, Programmer |  |

1. ***Work breakdown structure***
   1. **Work Breakdown Structure:**

* 1. **Schedule and Milestones:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Milestones | Description | Milestone Criteria | Planned Date | Priority |
| M0 | Planning | Recruit 4 members  Choose project | 11/9/2017 – 17/9/2017 | 1 |
| M1 | Handle requirement | Obtain requirement  Analyze requirement  Design the draft of main function | 17/9/2017 – 1/10/2017 |  |
| M2 | Start Project | Decide objectives and scopes  Choose process model | 1/10/2017- 8/10/2017 | 1 |
| M3 | Prepare necessary software/ hardware | Prepare laptop  Install software  Test software | 8/10/2017 – 15/10/2017 | 1 |
| M4 | Design diagram | Design Use case, activity, sequence class diagram and ERD | 15/10/2017 – 29/10/2017 | 1 |
| M5 | Design database | Design database based on M­4 | 29/10/2017 – 5/11/2017 | 1 |
| M6 | Design front-end | Design all pages’ view  Add new techniques from web’s course  Initiate back-end process | 5/10/2017 –12/11/2017 | 2 |
| M7 | Design back-end | Create functions based on M4 | 12/10/2017 – 26/11/2017 | 2 |
| M8 | Verify front-end + back-end | Test and debug to guarantee all function in front-end and back-end synchronized | 26/10/2017 – 5/12/2017 | 2 |
| M9 | Write report |  | 12/10/2017 – 5/12/2017 | 3 |
| M10 | Rehearse project |  | 8/12/2017 | 3 |
| M11 | Present project |  | 10/12/2017 | 3 |

|  |  |
| --- | --- |
| Backlog item | Estimate (day) |
| Allow member to watch movie, create/edit profile | 7 |
| Allow administrator to edit account, manage movie | 7 |
| Allow administrator to upload links, movie, details | 7 |
| Allow user to search movie | 7 |
| Total: | 28= 4 sprints(4 weeks) |

* 1. **Budget**

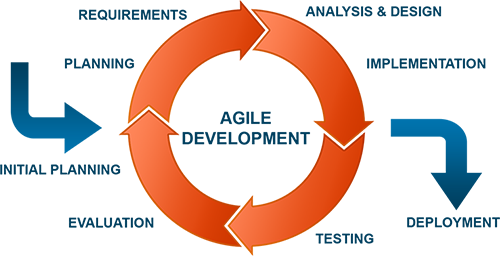
|  |  |  |
| --- | --- | --- |
| **Category** | **Budget** | |
| **Buy once** | **Pay per month** |
| Human resources |  | 200$ |
| Hardware | | |
| Mainboard | 120$ |  |
| CPU | 150$ |  |
| RAM | 50$ |  |
| HDD/SDD | 100$ |  |
| Power supply | 50$ |  |
| Case | 20$ |  |
| Monitor | 120$ |  |
| Mouse + Keyboard | 30$ |  |
| Printer | 200$ |  |
| Server (cloud) |  | 0$(Tomcat virtual server) |
| Software | | |
| MS Window 10 | 300$ |  |
| MS Office 2016 |  |
| MS Visio |  |
| Navicat |  |
| Domain + Host |  | 15$ |
| Total | 1140$+15$/month | |

* 1. **Development Process:**

We apply increment model in this project due to these reasons:

* We can plan, implement, design, test and debug incrementally
* After every component is accomplished, they could be sent to clients
* The clients can make feedbacks and our team can fix these bugs as soon as possible
* The outcome is able to easily meet the clients’ demand and maintain

Our method can be described like this:



* 1. **Development Environment:**

1. HTML5: to create an outline of the web pages
2. CSS + Bootstrap + Jquery + TinyMCE: to design the looks of the web pages
3. JavaScript: to create animations for the webpages
4. Java Server Page, Servlet: keep website up-to-date, create dynamic web pages
5. JavaBean: encapsulate many objects into a single object, and use it in website
6. Cookie: auto login for user if user request.
7. Mysql, Mysql Workbench: design and Store database of the system
8. Netbean: develop the project
9. Tomcat: server for running and testing website.
10. Visio: draw graph, Use Case, Class Diagram, Sequece Diagram, Activity Diagram, ERD, ….

MVC design pattern is applied to implement the system:



MVC is a software architecture - the structure of the system - that separates domain/application/business (whatever you prefer) logic from the rest of the user interface. It does this by separating the application into three parts: the model, the view, and the controller.

Model: Mapping Object from database to create and interact between Database and Controller

View: Graphic User Interface, interact with controller to get information.

Controller: Control database access, dataflow, update website whenever data changes, contain algorithms. Controller separate data and view.

* 1. **Measurement Program**

We plan to use these sites and methods to measure our website’s performance:

* https://tools.pingdom.com/ : test ping, load time, content size,…
* http://responsivetest.net/#u=http://www.zootemplate.com|1024|768|1 : test front-end in different platforms
* Use command line to check ddos attack
* http://www.alexa.com/ : check website’s ranking

1. ***Risk Managerment:*** 
   1. **Project risk:**

|  |  |  |
| --- | --- | --- |
| **Risk factor** | **Risk Impact Description** | **Mitigation &Contingencies** |
| Members don’t have coding skill and knowledge | Waste time to finding way to work.  Waste time to correct. | Apply a training session. |
| Member don’t have enough time to meeting and work together. | Hard to share and manage work | Use Skype, Facbook chat or Zalo to host a meeting online |
| Some members have a serious health problem | Work will be delayed until a member comeback | Share work to another member when the health problem occur |
| Work take more times than predicted | Can’t finish the work in time | Decide deadline carefully and have some solution if the risk occur |

* 1. **Product risk:**

|  |  |  |
| --- | --- | --- |
| **Risk factor** | **Risk Impact Description** | **Mitigation & Contigencies** |
| Lack of money | Can’t pay for maintaining server, buying new hardware | Save money to deal with unexpected situations |
| Unknow bug | Project can’t run, delayed until find the bug | Test the project carefully |
| Virus/hacker | Data will be lost or damaged. | Use anti-virus software, have some security soltutions, backup data |
| Choose old version of software | Project isn’t meet requirement  Have some unknow bug | Save money to buy new version of software if necessary |

* 1. **Bussiness risk:**

|  |  |  |
| --- | --- | --- |
| **Risk factor** | **Risk Impact Description** | **Mitigation & Contigencies** |
| Don’t attractive enough for customer | Can’t continue to maintain and develop project | Make plan to improve the quality of the result. |
| Marketing project is not effective | Don’t have enough customer to retrieve profit | Hire some marketing expert. |
| Can’t find the investor to raise fund | Lack of budget | Try to find the investor by apply some TV show. |

1. ***Delivery Plan:***

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Development phase** | **Alpha-test phase** | **Close-beta phase** | **Open-beta phase** | **Commercial release** |
| **Description** | Our team develop, test project | Invite small amount of users use system and collect feedback | Increase amount of users, continue to collect feedback, track bandwith, response time | Allow everyone use system, continue to collect feedback, track bandwith, response time | Complete fix bug, adjust system. Focus on maintaining system |
| **Time** | 11/9/2017- 10/12/2017 | 23/12/2017-  15/1/2017 | 20/1/2017- 24/2/2017 | 28/2/2017- 6/5/2017 | From 15/5/2017 |

**Sprint backlog:**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **1 st sprint( Design general GUI)** | | | | | | | |
| **Task** | **Monday** | **Tuesday** | **Wednesday** | **Thursday** | **Friday** | **Saturday** | **Sunday** |
| Define UC, Requirements | 2h |  |  |  |  |  |  |
| Define function | 2h |  |  |  |  |  |  |
| Draw ERD | 2h | 2h |  |  |  |  |  |
| Define testcase | 2h | 2h |  |  |  |  |  |
| Define entities, relation sets |  | 1h |  |  |  |  |  |
| Insert data to database |  | 2h | 3h | 3h |  |  |  |
| Design pages |  |  | 2h |  |  |  |  |
| Design interface for function |  |  |  | 1h | 2h |  |  |
| Validate pages |  |  |  |  | 2h | 2h | 3h |
| Test |  |  |  |  |  | 2h | 1h |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **2 nd sprint( Design general GUI)** | | | | | | | |
| **Task** | **Monday** | **Tuesday** | **Wednesday** | **Thursday** | **Friday** | **Saturday** | **Sunday** |
| Define UC, Requirements | 2h |  |  |  |  |  |  |
| Define function | 2h |  |  |  |  |  |  |
| Draw ERD |  | 3h |  |  |  |  |  |
| Define testcase |  | 2h |  |  |  |  |  |
| Define entities, relation sets |  | 1h |  |  |  |  |  |
| Insert data to database |  |  | 3h |  |  |  |  |
| Design pages |  |  | 2h |  |  |  |  |
| Design interface for function |  |  |  | 3h | 2h |  |  |
| Validate pages |  |  |  |  |  | 1h |  |
| Test |  |  |  |  |  | 2h | 1h |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **3 rd sprint( Design general GUI)** | | | | | | | |
| **Task** | **Monday** | **Tuesday** | **Wednesday** | **Thursday** | **Friday** | **Saturday** | **Sunday** |
| Define UC, Requirements | 4h |  |  |  |  |  |  |
| Define function | 1h |  |  |  |  |  |  |
| Draw ERD |  | 1h |  |  |  |  |  |
| Define testcase |  | 1h |  |  |  |  |  |
| Define entities, relation sets |  | 3h |  |  |  |  |  |
| Insert data to database |  | 2h |  |  |  |  |  |
| Design pages |  |  | 2h |  |  |  |  |
| Design interface for function |  |  |  | 2h |  |  |  |
| Validate pages |  |  |  |  | 2h |  |  |
| Test |  |  |  |  |  | 2h | 1h |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **4 th sprint( Design general GUI)** | | | | | | | |
| **Task** | **Monday** | **Tuesday** | **Wednesday** | **Thursday** | **Friday** | **Saturday** | **Sunday** |
| Define UC, Requirements | 1h |  |  |  |  |  |  |
| Define function | 2h |  |  |  |  |  |  |
| Draw ERD |  | 2h |  |  |  |  |  |
| Define testcase |  | 2h |  |  |  |  |  |
| Define entities, relation sets |  | 3h |  |  |  |  |  |
| Insert data to database |  |  | 1h |  |  |  |  |
| Design pages |  |  | 2h |  |  |  |  |
| Design interface for function |  |  |  | 1h | 2h |  |  |
| Validate pages |  |  |  |  | 2h | 2h |  |
| Test |  |  |  |  |  | 2h | 1h |

1. ***Abbreviations and Definitions***

|  |  |
| --- | --- |
| **Word** | **Meaning** |
| Java, C/C++ | Programming language |
| Data Structure, Dynamic Programming, Graph | Type of programming problems |
| Linux/Unix | Name of operating systems |
| Database | Organized collection of data |
| GUI | Graphical user interface: interface allow users interact with system |
| Front-end | Interface between user and back-end |
| Back-end | is basically how the site works, updates and changes. This refers to everything the user can’t see in the browser, like databases and servers. |
| Bug | an error, flaw, failure or fault in a computer program or system that causes it to produce an incorrect or unexpected result, or to behave in unintended ways |

1. ***References***