**Movie Database:** Feasibility Study

**Project Overview:**

The goal of this project is to develop a movie database that anyone can search through and, if an account is created, add content to. It will be primarily a desktop PC application, with the potential to be ported to mobile systems such as Android or Apples IOS. This application will be presented to the user as a neat graphical user interface with a database back end containing the content. An executable of the application will be available for download with no install required.

**Benefits:**

This application will provide a valuable community for individuals that are interested in movies to interact with each other and share views and opinions. They will also be able to add profiles of movies to the database for others to view. With user submitted content, this community would have the potential to grow, which in turn would attract more users to help it grow further.

The app will also provide a place for people in the movie business to come and see reviews of their works, which will in turn, allow them to find out why something was or wasn't well received. They may also add upcoming titles to the database for advertising purposes.

If the community becomes significant enough it may entice people in the movie business to request a partnership with the community, allowing them to possibly advertise their upcoming works or get exclusive reviews for current works. This would also be valuable for the reputation of the community as the presence of some esteemed company would add a degree of credibility to the whole project.

**Scope:**

The application will be designed for any platform that can run Java applications. It will be tested on PC, Mac OS and Linux with intent to develop for the Android and IOS mobile markets in the future. The application will not be available on the Playstation or Xbox platforms. A database will be stored in the cloud to house all of the content generated by the application users and the Java GUI front end of the application will be downloaded onto the users device to provide a client interface for database access.

**Walk-Through Scenarios:**

*User:*

Users will run the executable and be presented with the option to login, create an account or search through movies. An account will not be required to search and view movie profiles, one will be required however, to add movie profiles, add reviews for movies and flag other peoples content. Users will not be granted access to an account unless they provide the correct login details.

*System Administrator:*

The system admin will be required to login for his/her permissions to be active. Once logged in he/she will have the all of the permissions that the user has, in addition he/she will also have permission to search users, take down flagged content, remove a users account, reset a users password and he/she will have full access to the content on the database. The system admin will not be granted access unless the correct login details are provided.

**Preliminary Software Requirements Analysis:**

***Functional Requirements:***

*End User:*

* Search: The user should be able to search through the movies on the database without having to login.
* Create Account: The user should be able to create an account for free. A space in the database will be allocated for account information.
* Login: Once an account is created the user should be able to log into the account, assuming the correct login details are provided.
* Create Movie Profile: While logged in the user should have the ability to create a movie profile, the movie profile will be stored in the database.
* Add Review: While logged in, the user should have the ability to ad a review to any movie they want, the reviews will be stored in the database.
* Comment on Reviews: While logged in the user should be able to comment on other peoples reviews, comments will be also be stored in the database.
* Flag Content: While logged in, the user should have the ability to flag any content that the deem to be false, misleading or derogatory. A notification will be sent to the system administrator when anything is flagged.

*System Administrator:*

* Login: The system admin should be able to log in, an account will be created for the system admin when construction of the software is complete. System admin must be logged in to have system admin permissions.
* Create Movie Profile: The admin should be able to create a profile for a movie, admins movie profiles will be stored in the same place as users movie profiles.
* Add Reviews: The admin should be able to add reviews to movies, these reviews will appear the same as users reviews, only there will be and icon indicating the admin posted the review.
* Comment: The admin should be able to comment on reviews, admin comments will also have an icon indicating the comment is from the admin.
* Delete Content: The admin should be able to delete any flagged content, such as, movie profiles, reviews and comments. The user who posted the content will get a notification when something they posted is deleted.
* Delete Users Accounts: The admin should be able to delete users accounts if they continuously post false or inappropriate content. Unless the user is ip banned they will be able to create another account.
* Search: The admin should be able to search through both movie profiles and users.

*System:*

* The system should be able to compare entered login details with those in the database and perform an action based on the comparison results.
* The system should perform a searching algorithm to yield search results when something is entered into the search field.
* The system should be able to write information to the database; Account details, movie profiles, reviews and comments.
* The system should be able to process the users interaction with the GUI, such as button clicks, resizing, maximising, minimising etc..
* A notification should appear on screen if some internal error has occurred.
* The system should alert a user if they are creating a movie profile that already exists.
* The system should be able to prevent people from creating duplicate user names.

***Non-functional Requirements:***

* The system should process requests in under 10ms.
* The system should carry out searches in a reasonable amount of time.
* The system should be easily usable for any person.

***Technical Requirements and Feasibility:***

The system will be modelled using the unified modelling language and it will be documented using Rational Rose. This software is readily available.

The language used to implement this application will be Java SE using JDK 1.7. The GUI will be implemented using Swing and AWT in Java. This is readily available.

All of the content and user accounts will be stored in an online database. This database will be developed using mySQL Workbench, which is readily available.

**Deliverables:**

***Management Deliverables:***

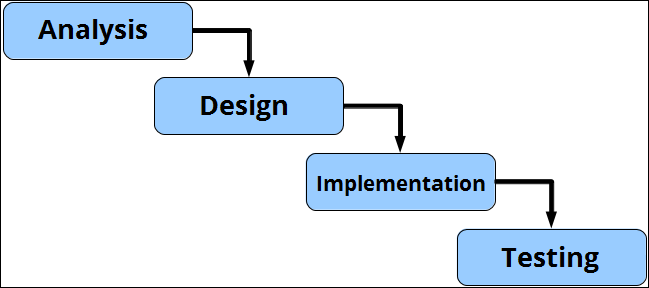
1. *Feasibility Study:* A Feasibility Study will be carried out to determine if this project can be successfully completed. This study will walk through the requirements, risks, scope and benefits of the project.
2. *Analysis and Design Plans:* Analysis and design will be carried out once the project is deemed feasible, this will layout the details for the project. This will include the software model, a plan for the logic of the app, a design plan for the GUI and a design plan for the database.
3. *Source Code:* When the Analysis and Design phase has been completed, the code for the software will be written, along with all related documentation. The design plans from the previous stage will be used as blueprints.
4. *Testing Documentation:* Once the implementation stage is complete, the software will be thoroughly tested for usability and bugs, these tests and test results will be documented.

***Technical Deliverables:***

1. A Graphical User Interface for both users and the system admin to interact with the app.
2. A database to store all content and account details.
3. The logic that will allow the GUI to interact with the database.

**Software Development Process:**

A standard waterfall model will be used as the software process model. We chose this model because we understand the system requirements. We are confident that we can complete each part of the development process fully before moving on to the next. We can clearly see the order in which the parts of the development process should be executed. See Figure 1 for image reference.

Figure 1: Waterfall Model

**Risk Analysis:**

*Java Runtime Environment Problems:*

The JRE may not be present on the clients device, resulting in the application not being able to run on said device.

*Solution:*

We must specify that the Java Runtime Environment is required to be able to use the application on any device.

*Limited Knowledge of Databases:*

The groups collective knowledge of databases is somewhat limited as little time as been spent working on them.

*Solution:*

There will be a significant amount of time spent designing the database, a significant amount of time will also be spent researching the tools needed to develop the database.

*Little to no users:*

When the application is launched there will be no community, since the content of the application is mainly user generated, there will be very little content contained in the database.

*Solution:*

The group will spend time building up some content to kick start the project, this will attract other users to the application and thus, build up the community.

*Time:*

The project will take quite some time to complete if the analysis and design phases of development are not done correctly.

*Solution:*

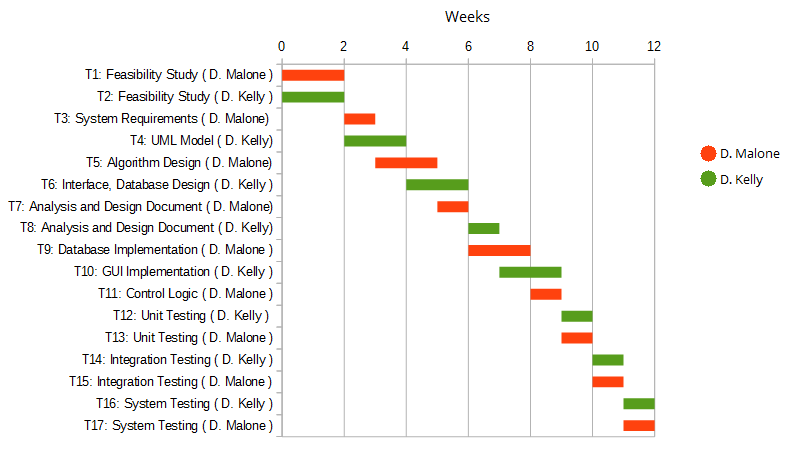
Spend the correct amount of time on the analysis and design phases and keep up persistent work so as not to get lazy.

**Visibility Plan:**

The group will consult in a lab on Tuesday from 11am – 1pm every week to discuss project related issues and ideas. The meetings will be documented, notes will be taken of any problems that may arise. The group will also keep in contact over Skype to share ideas and help each other with any issues.

**Project Plan:**

The project has been scheduled using a gantt chart which can be seen below. It specifies the tasks that each group member will be carrying out in order to complete the project and the time line in which each task will be done.



**Conclusion:**

We have concluded that this project is feasible, and it will go ahead. The project is subject to small changes if necessary once the analysis and design phases begin. The final application and documentation will be submitted in May 2014.