CA400 Final Year Project: Functional Specification

Predictive Analytics Platform

Sean Quinn 13330146

25/11/16

1. Introduction

1.1 Overview

The aim of this project is to create a predictive analytics platform which will provide the user with a suite of customisable classification algorithms. This interactive application will have a graphical user interface which will provide a fast and user friendly way of configuring, optimising and running the algorithms as well as presenting the results in both graphical and numerical formats. Classification algorithms are at the core of predictive analytics and in order to maximise the value provided by the system I have chosen four of the most widely adopted prediction algorithms. The K-Nearest Neighbours, Decision Tree, Naive Bayes and Support Vector Machine classification algorithms. The combination of these four algorithms along with a user friendly interface and aesthetic visual outputs will provide the user with a powerful tool for making predictions based on their data.

secondary to the creation of this tool I intend to include a research element to this project. I have obtained a comprehensive dataset which contains thousands of metrics for hundreds of countries from the years 1960 to 2015. I will analyse the economic data contained in this dataset using the analytics platform and I will attempt to discover some unique insights as to what patterns emerge in the data in the years prior to a recession with the aim of trying to predict the likelihood of a country entering into a recession. This will involve applying predictive analytics techniques to this extremely complex economics problem. This aspect of the project will also act as a stress test and an example of a practical application for the analytics platform.
link to dataset:
https://www.kaggle.com/worldbank/world-development-indicators

1.2 Business Context

This system will be designed as a standalone application which could potentially be used by any business or organisation which seeks to investigate its data with the aim of mining patterns and insights and making predictions. It would allow an organisation to make use of these complex algorithms and take advantage of their results without needing to have a data scientist or somebody with a deep understanding of data analytics in house. It will bring advanced analytics to the non-expert analyst. As such it may be of value to students or smaller organisations where there is not an individual whose role in the organisation revolves solely around data analytics. I feel that any measure of success achieved as a result of my efforts to use analytics to predict economic recession may be of significance as this is a problem of particular importance to governments and economists.

1.3 Glossary

- **Data Mining:**
  Data mining is the computational process of discovering patterns in large data sets involving methods at the intersection of artificial intelligence, machine learning, statistics, and database systems. The overall goal of the data mining process is to extract information from a data set and transform it into an understandable structure for further use.
- **Predictive Analytics:**
  Predictive analytics is the branch of advanced analytics which is used to make predictions about unknown future events. Predictive analytics uses techniques to analyze current data to make predictions about the nature of future data.
- **API:**
  Application Program Interface - is a set of subroutine definitions, protocols or tools for building software and applications. It is used to specify the interface between two pieces of software and manage how they interact.
- **REST:**
  REST is a type of web service where requests made to a resource's URI will elicit a response in XML, HTML, JSON or some other defined format. By making use of a stateless protocol and standard operations REST systems aim for fast performance, reliability, and the ability to grow, by using reused components that can be managed and updated without affecting the system as a whole, even while it is running.
- **GUI:**
  Graphical User Interface - is a type of user interface that allows users to interact with electronic systems through graphical icons and visual indicators such as secondary notation, instead of text-based user interfaces, typed command labels or text navigation.
- **K-Nearest Neighbours (KNN) Algorithm:**
  K-Nearest Neighbours is a type of instance-based learning, or lazy learning, where the function is only approximated locally and all computation is deferred until classification.
- **Decision Trees (DT) Algorithm:**
  Decision tree learning uses a decision tree as a predictive model which maps observations about an item (represented in the branches) to conclusions about the item's target value (represented in the leaves).
- **Naïve Bayes (NB):**
  Naive Bayes classifiers are simple probabilistic classifiers based on applying Bayes' theorem with strong (naive) independence assumptions between the features.
- **Support vector machine (SVM):**
  Support vector machines are supervised learning models with associated learning algorithms that analyze data used for classification

and regression analysis. Given a set of training examples, each marked as belonging to one or the other of two categories, an SVM training algorithm builds a model that assigns new examples to one category or the other, making it a non-probabilistic binary linear classifier.

2. General Description

2.1 Product / System Functions

This system is designed to allow the user to run the algorithms with as little user effort as possible. So as such the graphical user interface strives to be as simplistic and user friendly as possible. The application will display four tabs which represent the four different algorithms available to the user. Upon selecting any of the tabs the user will be presented with a screen where they must enter in the relevant parameters and configurations required to run the algorithm. This includes specifying the dataset and selecting parameters specific to the algorithm. For example when running the K-nearest neighbours algorithm the user will be asked to specify the location of the training set and validation set, or specify one dataset and select a tickbox to ask the application to partition this set into training and validation sets automatically, select how many neighbours to use (the value of k), enter the index of the column which represents the class attribute of the training set, select True/ False as to whether there is a header row present in the dataset, select which distance algorithm the KNN algorithm will use in its calculations, select if the distance algorithm should be weighted and with what weighting algorithm. Once all of that has been selected the user can select run. Each of the four different algorithms will have different fields to be selected on their respective selection pages to allow the algorithm to run. When these fields have been filled in the application will allow the user to run the algorithm and it will then present a results screen which will contain the numerical and graphical outputs of running the algorithm. On the results page the user will have a reset button available to bring them back to the selection page and a button to export the results to a PDF. The selection screen will contain an information icon which when selected will provide further information about the fields to be entered to assist the user. Similarly the outputs screen will contain an information icon to further explain the outputs and graphs of the algorithm. At all times there will be an exit button which will close the application.

2.2 User Characteristics and Objectives

The intended user of the analysis application is anyone who wishes to conduct data mining or is involved in using predictive analytics techniques. The user will need to have some basic knowledge of data preprocessing in order to ensure the dataset provided to the application is in an appropriate format for the algorithms to run on it. The required format for the input data file will be clearly specified to the user in the GUI. There will be some preprocessing such as normalisation carried out by the algorithms but other dataset specific features of preprocessing such as data cleansing or data integration will not be carried out by the application. In order to take

advantage of this systems features fully the user must understand the nature of the data they are performing the analysis on. The intended user must understand the meaning of the outputs of these algorithms and they must have a problem or curiosity in relation to the data which can be satisfied by running these algorithms. This is intended to be used by someone with some knowledge of data analytics but does not require the user to be competent in programming or understand the inner workings of the algorithms.

2.3 Operational Scenarios (Use Cases)

2.3.1 Run Algorithm

Precondition: In order to execute the algorithm the user must have opened the application and filled in all the required details on the algorithms selection page. The user must have a dataset in an appropriate format to allow the algorithm to execute.
Main Flow of Events: The graphical user interface will take the information entered by the user and pass it to the algorithms engine which will execute the algorithm on the dataset provided in the user's arguments.
Postcondition: The user is presented with a results page showing the numerical and graphical outputs of this particular algorithm.

2.3.2 Export to PDF

Precondition: The user has already executed the algorithm and has been presented with a results page.
Main Flow of Events: The user selects the save button which allows the algorithms outputs to be exported to a pdf.
Postcondition: The output of the algorithm has been saved to a pdf file.

2.3.3 Reset Algorithm

Precondition: The user has executed an algorithm and is now presented with the results page.
Main Flow of Events: The user selects the reset button to navigate back to the argument selection page for this algorithm from the results page in order to re-configure.
Postcondition: The user has now been presented with a blank selection page and unless the user chose to save the output of the last execution to a file then it will have been discarded.

2.3.4 Get Help

Precondition: The user is on either the selection or results page for any algorithm and wants to get more information on what it is that they are seeing on the page.
Main Flow of Events: The user selects the information icon and is presented with a more detailed description of what they are seeing on the screen. If they are on the selection page they will get an explanation of each of the arguments to be entered, if they are on the results page they will get an explanation of the outputs and their meaning.

Postcondition: The user has now accessed the information page for this screen and the content of the page has been explained.

2.3.5 Switch Algorithm

Precondition: The user has opened the application and is on either the selection or results page for any of the algorithms.
Main Flow of Events: The user selects an alternative algorithm from the tabs across the top and is brought to the selection page for that algorithm or the results page if the algorithm has been run and not been reset.
Postcondition: The user has now changed algorithm.

2.4 Constraints

2.4.1 Time Constraints

The completion of this project will be time sensitive due to the fixed deadlines. While I am confident I can complete the analytics platform to a satisfactory standard, time constraints may have an impact on the scope of the project, as I cannot afford to compromise on quality. As a contingency plan I have also included a fallback desktop architecture in the event that I do not have time to implement the client server REST architecture. The analytics platform is of higher priority than the research aspects and so if cuts to the scope must be made they are likely to be made to the research.

2.4.2 Hardware Limitations

The running of the algorithms provided by this program will require significant processing power, but for development purposes the power available on my desktop machine will suffice.These algorithms have demanding time complexities and as such increasing the number of attributes in the input dataset will have a large effect on the time taken to run the algorithm with the processing power available to me.

2.4.3 Research Constraints

The completion of the research element of this project is constrained by the fact that I am attempting to accomplish something almost impossible in predicting the onset of a recession. Total success in this is impossible, but there is scope to gain insights and identify contributing factors. Ultimately I am constrained by the inherent complexity of the problem from ever achieving a holistic solution.

3. Functional Requirements

3.1 K-Nearest Neighbours Algorithm Engine

- **Description -** The k-nearest neighbours algorithm must be implemented in code and provide an API so calls to the engine can be made from the GUI.

- **Criticality -** This is one of the four core algorithms of the system so it is an essential function of the system.
- **Dependencies with other requirements -** In order to eliminate the dependency between this requirement and the GUI, it will be coded in such a way as to provide a GUI independant API, so that the algorithm can be easily paired with any Interface.

3.2 Decision Tree Algorithm Engine

- **Description -** The decision tree algorithm must be implemented in code and provide an API so calls to the engine can be made from the GUI.
- **Criticality -** This is one of the four core algorithms of the system so it is an essential function of the system.
- **Dependencies with other requirements -** In order to eliminate the dependency between this requirement and the GUI, it will be coded in such a way as to provide a GUI independant API, so that the algorithm can be easily paired with any Interface.

3.3 Naive Bayes Algorithm Engine

- **Description -** The naive bayes algorithm must be implemented in code and provide an API so calls to the engine can be made from the GUI.
- **Criticality -** This is one of the four core algorithms of the system so it is an essential function of the system.
- **Dependencies with other requirements -** In order to eliminate the dependency between this requirement and the GUI, it will be coded in such a way as to provide a GUI independant API, so that the algorithm can be easily paired with any Interface.

3.4 Support Vector Machine Algorithm Engine

- **Description -** The support vector machine algorithm must be implemented in code and provide an API so calls to the engine can be made from the GUI.
- **Criticality -** This is one of the four core algorithms of the system so it is an essential function of the system.
- **Dependencies with other requirements -** In order to eliminate the dependency between this requirement and the GUI, it will be coded in such a way as to provide a GUI independant API, so that the algorithm can be easily paired with any Interface.

3.5 Graphical User Interface

- **Description -** The graphical user interface needs to be created in a manner that enables the users execution of the algorithms. The user interface will have to contain several different types of graphs and

graph components. It will have to be user friendly and as aesthetically pleasing as possible.
- **Criticality -** The user interface is essential to the success of the system.
- **Dependencies with other requirements -** The graphical user interface will have a dependency on the algorithm engines, as it will be coded to adhere to their respective API's.

3.6 Export to PDF

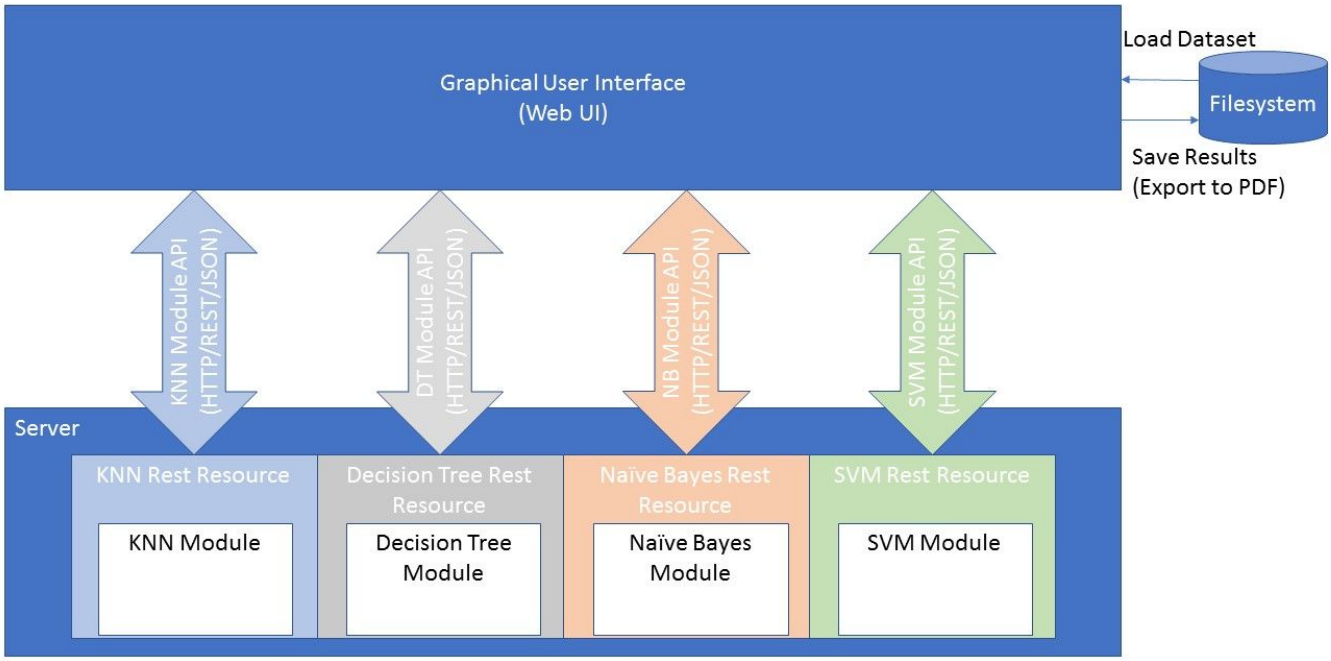
- **Description -** This function will need to take all the outputs presented to the user in the outputs screen of a given algorithm and export them to a PDF format.
- **Criticality -** This requirement is of medium criticality as the system will still work without it but the inclusion of this feature is preferable to screenshoting the application to save the results.
- **Dependencies with other requirements -** There is a dependency between this requirement and the graphical user interface.

3.7 Research Into Economic Recession

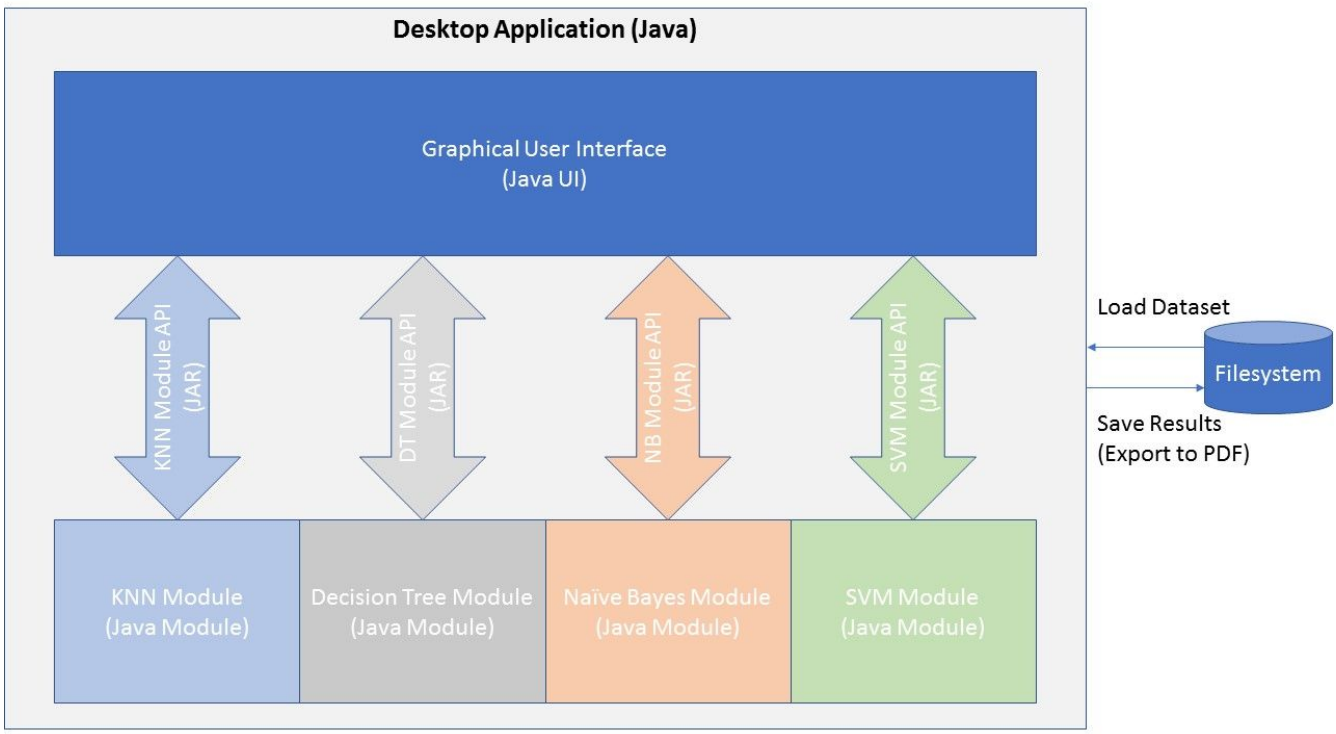
- **Description -** This is the analysis which will be carried out on the dataset I have obtained using the analytics platform and other methods.
- **Criticality -** This is an important element of the project, as it was a large part of the reason for creating the system, but as it is not critical to the success of the analytics platform it is of secondary priority compared to the other requirements.

4. System Architecture

4.1 Architecture Diagram



4.2 Contingency Architecture Diagram

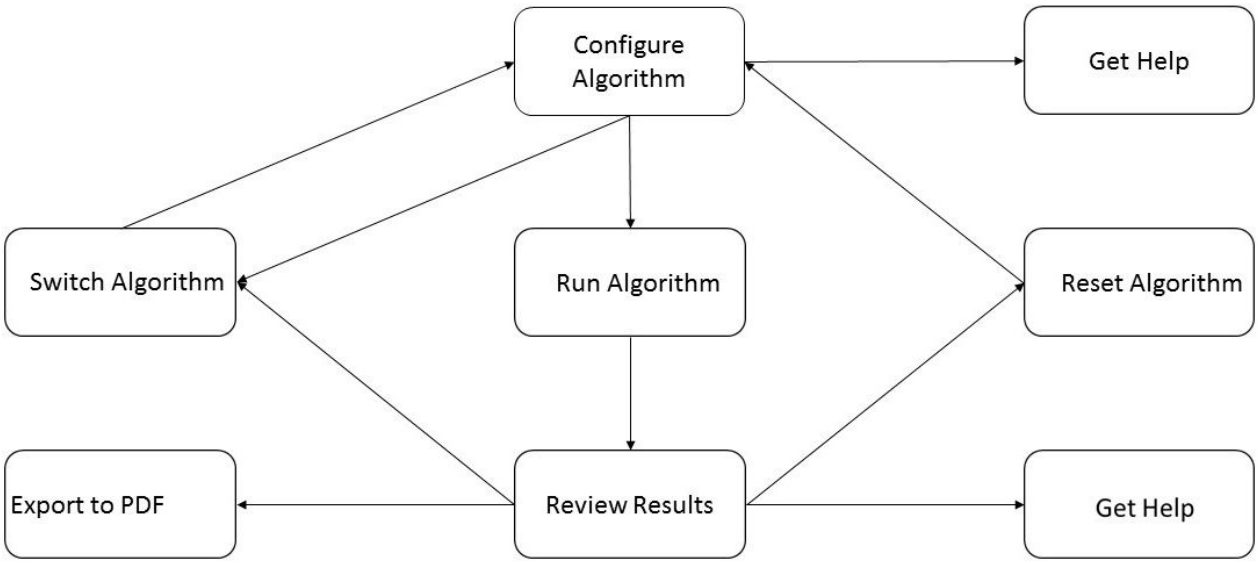


4.3 Architecture Overview

The proposed architecture above illustrates how I intend to structure the application. I will code four independent algorithm engines with their own respective APIs. This will allow me to implement a REST client-server architecture and code the GUI using web technologies. For the development phase the algorithms can be hosted locally on a tomcat server with the option of deploying to a web server on completion of the project. Another advantage of this architecture is that there is the option of deploying to multiple servers and distributing the job out to different machines, even running algorithms in parallel, which is something I would consider should the project run ahead of schedule. I have also included a contingency architecture diagram in the event that I need to reduce scope based on time constraints, this change would not affect the algorithms should I choose to make it as the same API's could be utilised.

5. High-Level Design

5.1 High Level Design Diagram



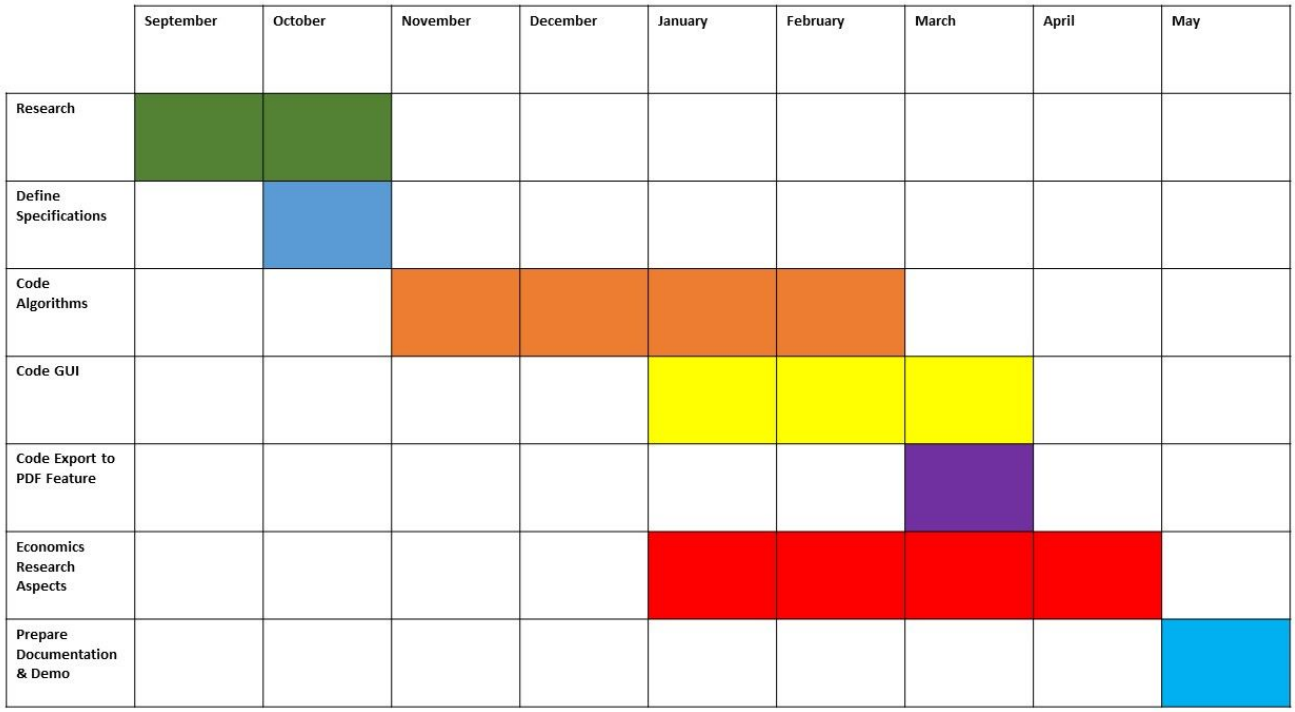
5.2 High Level Design Description

The above diagram illustrates the high level design of the systems functionality. At any point in the systems operation the user will be using one of the eight functions above. When the program is started the user will be brought to a configuration page for one of the algorithms. From this page they can use the get help function for assistance with the algorithms run parameters, they can run the algorithm once the parameters have been filled in or they can switch algorithm and be brought to the configuration page for a different algorithm. Once the algorithm has finished running the user will be

brought to the review results screen where the can see the graphical and visual outputs of the algorithm. From this screen they can access a get help function to explain the nature of the results, they can export the results of this run to a pdf, they can reset the algorithm to be brought back to the configuration screen or they can switch algorithm to be brought to the configuration screen of a different algorithm.

6. Preliminary Schedule

6.1 Gantt Chart



6.2 Schedule

The current project schedule as it stands currently in November is as shown in the above Gantt chart. The research and specification phases of the project have been completed and I have began coding the algorithms. By January I hope to be in a position to start the coding of the GUI and begin the research element of the project. I hope to code the export to pdf feature in March when the GUI is near completion and have all development finished by the end of April leaving May solely for documentation and preparation of demos and deliverables.