Table of Contents

1. Introduction

1.1 Overview

The goal of this project is to develop a **mobile interface** for the existing DCU **Machine Translation (MT)** System. The strategy is to develop this within the context of a prototype that can be used with other software applications developed within **CNGL**, the developers of this system.

With this interface, the user will be able to input a sentence or a paragraph in a language and have it translated into a desired language. Despite the high quality of the system, some translations may need further **post-editing** by human translators in order to achieve publishable quality. The interface will also offer the functionality for the user to post-edit the MT translations.

The current system allows for this, but on a desktop machine. The system is a **machine learning** based system, which uses **statistical models**. It automatically learns to translate from previously translated texts (**parallel corpus**) using statistical machine learning techniques. This is one of the most successful and popular approaches for machine translation.

The most important part of this project is portability. The **framework** must be re-usable with other work done in CNGL. It must also target as large an audience as possible. Obviously there are many challenges with this. Most importantly the design of the interface must not suffer for the sake of portability. Finding the correct balance will be the most difficult part of this project.

1.2 Business Context

Smartphones are predicted to sell over two billion units by 2015. Obviously this an important market to tap. A crucial decision is choosing the correct mobile device to target. As of November 2010 the statistics put Symbian as the market leaders in mobile operating systems, with a market share of 44.6%. This is significantly more than any of it's rivals; however by choosing Symbian you immediately miss out on 55% of the market. This is far from ideal, and due to time constraints developing this interface for extra platforms in each of their native languages is unrealistic. As a result it is being developed as a **web application**.

Machine Translation tools are plentiful. However, as the existing systems cannot be fully relied upon there is still room in the market for more. Furthermore, there is room for specialist machine translation systems that focus on specific[ᵻ] subjects. These can translate such documents with a much higher degree of accuracy. CNGL have such systems in place.

The success of this application relies heavily on the user experience. That will be the main focus of this project.

[ᵻ] From here forth specific denotes MT systems which focus on a specialist subject (law, travel documents etc)

1.3 Glossary

Mobile Interface: This is a graphical user interface built specifically for mobile devices.

Machine Translation (MT): This is the application of computers to the task of translating texts from one natural language to another.

CNGL: The Center for Next Generation Localisation - A dynamic Academia-Industry partnership with over 100 researchers developing novel technologies addressing the key localisation challenges of volume, access and personalisation.

Post-editing: The process of editing and modifying text after it has been compiled or translated by a machine.

Machine Learning: Machine learning refers to a system capable of the autonomous acquisition and integration of knowledge. This capacity to learn from experience, analytical observation, and other means, results in a system that can improve its own speed or performance, i.e., its efficiency and/or effectiveness.

Statistical Models: A statistical model is a mathematical representation of the relationship among variables.

Parallel Corpus: Texts that are translations of each other.

Framework: The basic structure of a database, process or program.

Web Application: A website that behaves like software.

IP Address: A unique number that identifies the precise location of a particular node on the Internet. The address is a 32-bit number usually written in dotted decimal format, i.e. in the form ‘123.33.22.32’.

SQL: A simple, commonly used, standard database programming language that is only used to create queries to retrieve data from the database.

URL with parameters: This is a IP or website address with information appended to the end.

eg. A user searches for Will Smith on Youtube -

http://www.youtube.com/results?search_query=***will+smith***&aq=f

2. General Description

2.1 Product / System Functions

Below is the basic functionalities of the MT system with the added mobile interface -

- Choose input language
- Choose desired output language
- Choose subject (General purpose, specific)
- Copy translation
- Email translation
- Post-edit translation

2.2 User Characteristics and Objectives

The user is a mobile user, but is not assumed to be overly "tech savvy". As such the most important thing is making the interface user-friendly, it must be simple and intuitive to use. Ideally the product would not be aimed at a small target audience. It should meet the needs of

- All smart phone users.
- A user using it to translate non-specific information (web-sites, emails etc).
- A user using to translate specific information.
- A business translating documents of a specific subject.

Of course meeting the needs of all the above is not practical. However the project shall strive to get as close as possible.

2.3 Operational Scenarios

<u>General Translation:</u>

The user starts by choosing the input language. The default will depend on their **IP Address**. Following this they choose the desired output language. This will initially be the same as the input language.

The user will then be presented with an input box into which they can enter text by typing, or by pasting from the clipboard. When ready, the user submits the text. The results appear in a new text box underneath from which the user can copy the text, or immediately send it as an email.

<u>Specific Translation:</u>

To make the translation more accurate for a particular document type the user can select the subject of the document. This can be changed in the settings (This may be changed to make it more accessible). Changing this any time before submitting text to be translated allows it to take effect.

Post-editing:

Following the translation the user can edit the results. The output can be edited similarly to inputting text, only the changes are made in the output text box. If the user makes changes they are presented with a submit button. Upon submission they are notified if successful. The post-edits are saved on the server.

To start the translation process again the user simply changes the inputted text and submits it again.

2.4 Constraints

Listed below are the constraints on this project -

Non-native Application:

Since the interface is being developed as a web application it cannot easily make use of each applications' native capabilities (gyroscope, accelerometer etc). It's performance will also suffer as it must work through the device's browser.

DCU Only:

The translation system is stored on a server in DCU. Unfortunately this server can only be accessed from within DCU. This will limit testing the system.

Number of Platforms:

Even though the interface is being developed as a web app, having it optimised for all mobile devices is unrealistic. This is largely due to screen sizes, with browser capabilities also being a factor. It will work best on widely used devices.

Testing:

Due to the sheer number of mobile devices available, testing it completely will not be possible.

Programming Language Limitations:

The majority of the interface functionalities will be programmed with javascript. As such, it will be limited to the capabilities of this language.

Data Sets:

The system will be limited by the number of data sets CNGL have. These are needed for each language, for general purpose translation, and for each specific area.

3. Functional Requirements

3.1 Mobile Interface Functionality

Description:

This is the main point of contact for the user, and the reason for the project. It should shield the user from the inner workings of the system, while providing them with a user-friendly and efficient work space.

Criticality:

As stated this is the purpose of the project. However it is not simply enough to provide the user with any usable interface. It must have all the necessary functionality, while being intuitive to use and visually appealing. The success of this project very much relies on the user experience. This is influenced majorly by the interface.

Technical Issues:

It may take some testing to optimise the interface. Once enough user feedback has been received the layout will be finalised.

Dependencies On Other Requirements:

This is the first and main point of contact for the user. Everything the user wishes to do must be done through this interface.

3.2 Translation

Description:

The user enters text. The input language must be chosen correctly or it will fail. The user also chooses the desired output language, and can indicate the subject of the input text if it is a listed specific subject. After submission they received the translated text.

Criticality:

Obviously this is necessary to the whole project, it is the principal functionality of the system. Without it you have an interface to nothing.

Technical Issues:

When the user submits their text for translation it is appended to the url for the server along with the other necessary information (input language, output language, subject). Any problems with the server will disable the translation functionality.

Dependencies On Other Requirements:

This depends on the interface being in place correctly to collect the needed information.

3.3 Storing Translation Information

Description:

When the user submits text for translation all information from the input and translated output will be stored in a text file or database on the server. Each entry will have a unique identifaction number.

Criticality:

This is not a crucial feature. However it will enable CNGL to gather information on the usage of the system. This should in turn help them improve it.

Technical Issues:

If a text file is used only one user may use they system at a time. Otherwise the integrity of the data will be compromised.

Dependencies On Other Requirements:

This depends on the translation stage.

3.4 Post-editing

Description:

After the user receives the translated text they may decide to edit the results. They do this by simply changing the text in the output box. When complete they submit their changes to the server. The changes are added to the information already stored in the database on the translation in question. This record is found using the translation's unique number.

Criticality:

This is not crucial to an MT system. Without it the translation functionality will operate correctly. However with time this will improve the quality of the translations returned.

Technical Issues:

The two big issues with post-editing is checking the quality of the changes, and storing them. Extra functionality for analysing these is being added which I discuss next. For the storage, depending on time constraints; either a simple text file with separators will be used, or a complete **SQL** database.

Dependencies On Other Requirements:

Like the translation stage, this depends on the interface being suitably implemented to facilitate post-editing. The translation functionality must also operate correctly.

3.5 Analysis of Post Editing

Description:

The amount of time a user spends editing the MT system's translation will be timed. The specific words and sentences they change will also be noted. These statistics will be stored with the post-editing, and will be used to analyse the quality of the changes.

Criticality:

This is crucial following the post-editing stage, otherwise changes may be submitted that weaken the future results of the system.

Technical Issues:

Certain assumptions will be made. For example – it's assumed the longer a user spends post editing, the higher the quality. This will not always be the case.

The analysis will take much fine tuning.

Dependencies On Other Requirements:

Without the post-editing functionality this addition is unnecessary.

3.6 Setting Default Language

Description:

When the user opens the application a default language will be set for the input and output. Both will be the language of the country they are in based on their IP address.

Criticality:

This is not a crucial feature, but it is simple to implement and should save the user time. This will not be useful until there is a full range of languages to choose from, and a significant number of people using the application.

Technical Issues:

The default languages may not be what the user is looking for.

Dependencies On Other Requirements:

None.

4. System Architecture

4.1 System Architecture Diagram

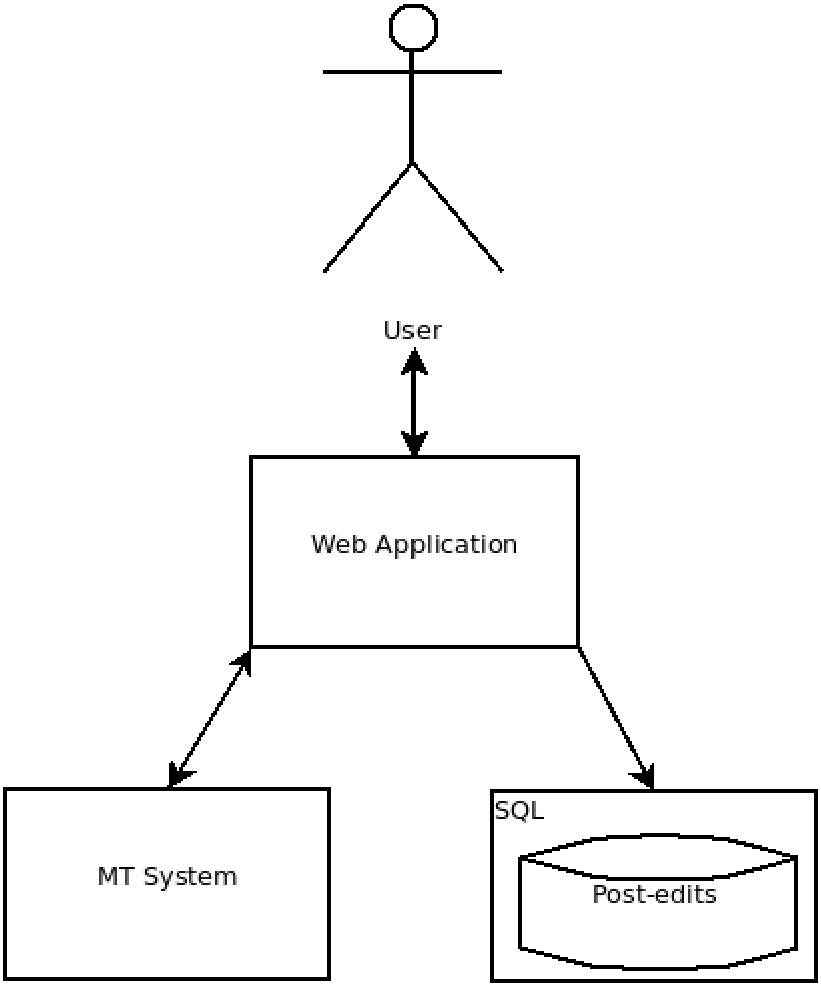


Fig 4.1

Fig 4.1 Depicts the architecture of the system being put in place. It shows clearly the 3 categories it can be separated into.

4.2 Web Application

This illustrates the interface the user will be presented with on their mobile device. It displays and makes usable all the functionality of the system while hiding the complexities from the user.

4.3 MT System

This element of the architecture represents the existing DCU MT system. It allows the user to submit text for translation. The text along with the input language, the desired output language, and the subject are appended to the IP Address of the server which hosts the system. This forms a **URL with parameters**. The system currently uses this to receive the information needed. Upon completion it returns a webpage with the result of the translation. A script will be used to fetch this, and it will be displayed to the user. All information from this translation will be stored on the server.

4.4 Post-edits

The application also deals with post-editing, and for design purposes this is considered a separate entity in the architecture to the translation. The post edits are stored in a database. This is designed in line with CNGL to ensure it can be used effectively to improve their system.

5. High-Level Design

5.1 High-Level Design Diagram

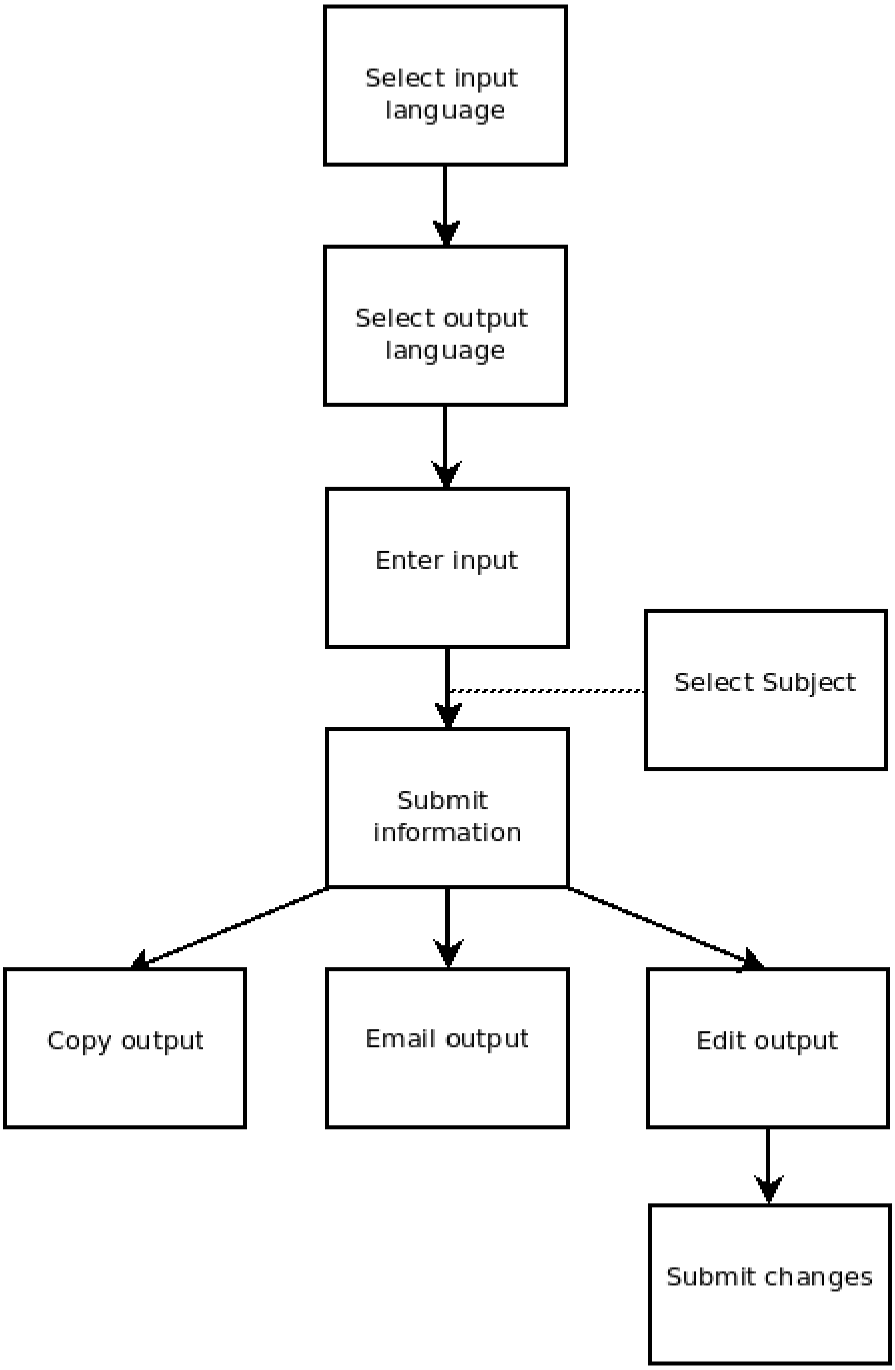


Fig 5.1

5.2 High-Level Design Description

Fig 5.1

- **Select input language :**

  Choose the language of the input text from a list of supported languages.

- **Select output language :**

  Choose the language of the output text from a list of supported languages.

- **Enter input :**

  Enter the text for translation in the input box. This can be typed, or pasted.

- **Select subject :**

  If the user chooses to they can specify the subject of the input text. This will increase the accuracy of the translation.

- **Submit information :**

  After the language selections have been made, and the text inputted, the user can submit the information for translation.

- **Copy output :**

  The user can copy the translated text to their clip board for use outside the application.

- **Email output :**

  The user can email the translated output. This will open their default mail application with the text in the body of the email.

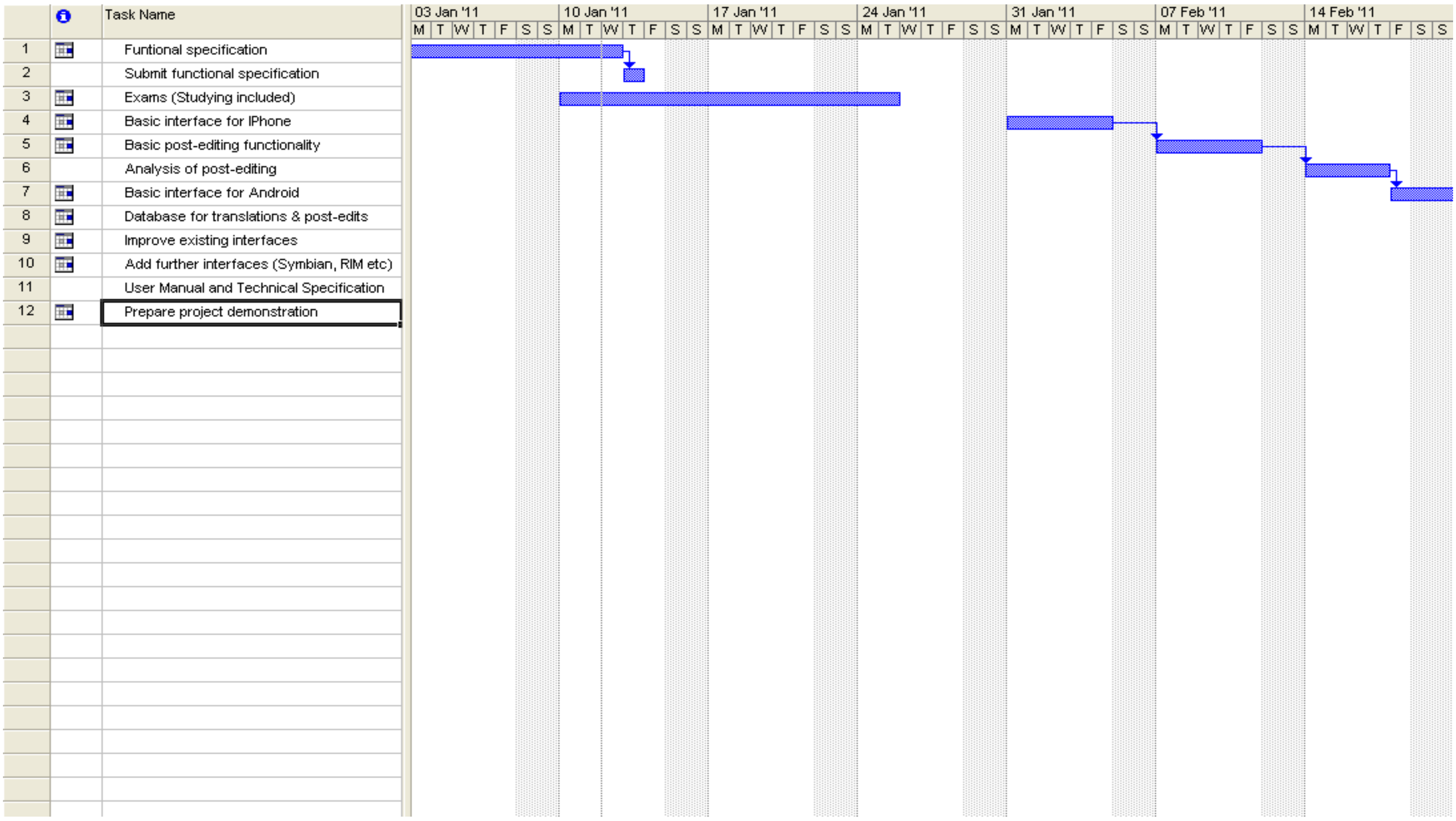
- **Edit output :**

  If the user chooses to they can edit the output.

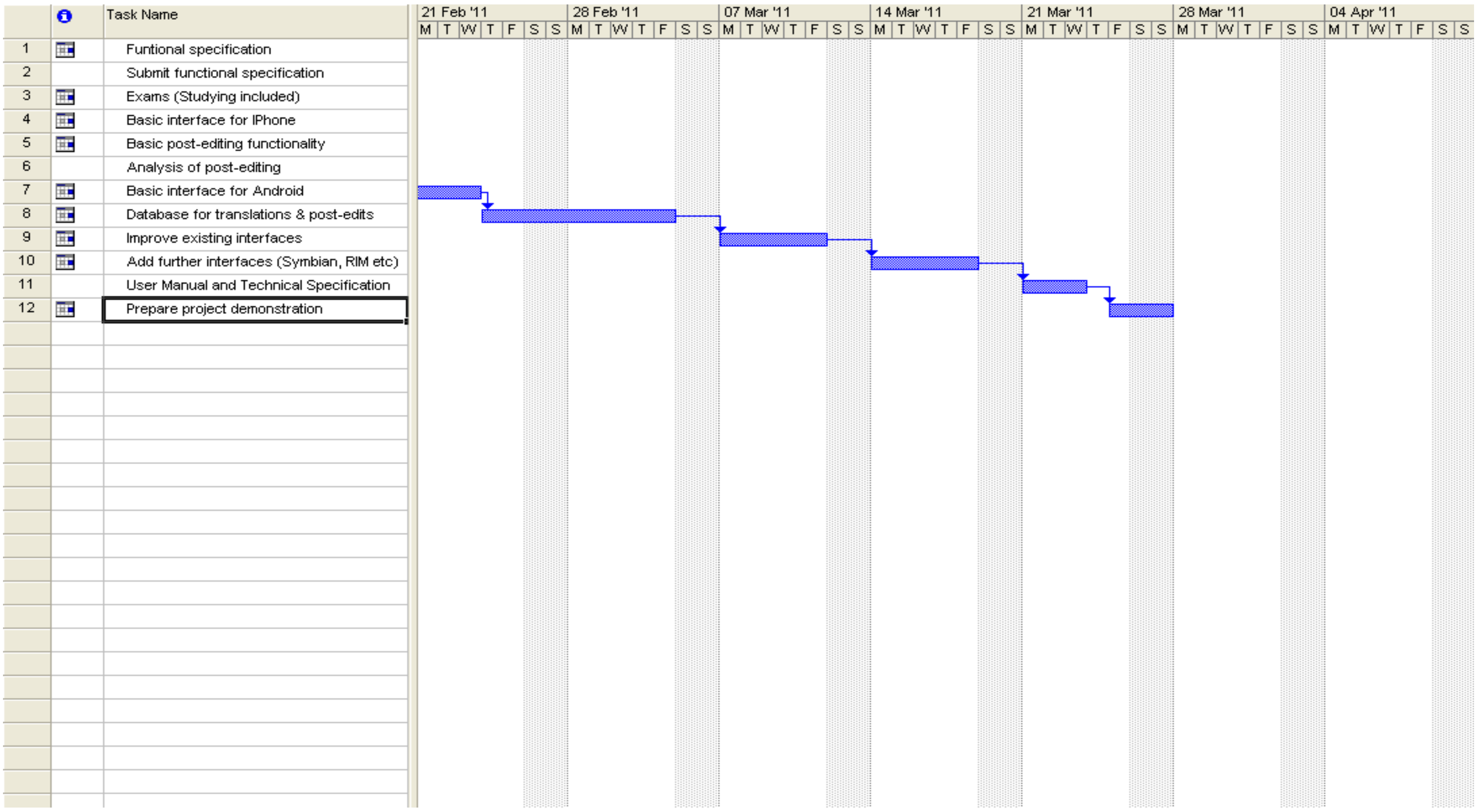
- **Submit changes :**

  The user can submit the changes to the server after editing the text.

6. Preliminary Schedule



	ℹ	Task Name
1	▦	Funtional specification
2		Submit functional specification
3	▦	Exams (Studying included)
4	▦	Basic interface for IPhone
5	▦	Basic post-editing functionality
6		Analysis of post-editing
7	▦	Basic interface for Android
8	▦	Database for translations & post-edits
9	▦	Improve existing interfaces
10	▦	Add further interfaces (Symbian, RIM etc)
11		User Manual and Technical Specification
12	▦	Prepare project demonstration



	ℹ	Task Name
1	▦	Funtional specification
2		Submit functional specification
3	▦	Exams (Studying included)
4	▦	Basic interface for IPhone
5	▦	Basic post-editing functionality
6		Analysis of post-editing
7	▦	Basic interface for Android
8	▦	Database for translations & post-edits
9	▦	Improve existing interfaces
10	▦	Add further interfaces (Symbian, RIM etc)
11		User Manual and Technical Specification
12	▦	Prepare project demonstration

7. References

- Coda Research Consultancy (May 2010) **Worldwide Smartphone Sales Forecast to 2015** [Internet] Research and Markets

  Available from: <*http://www.researchandmarkets.com/research/f80efc/worldwide_smartphone_sales_forecast_to_2015*>

  [Last accessed: 10/1/11]

- Gartner (November 2010) **Gartner Says Worldwide Mobile Phone Sales Grew 35 Percent in Third Quarter 2010; Smartphone Sales Increased 96 Percent** [Internet] Gartner

  Available from: <*http://www.gartner.com/it/page.jsp?id=1466313*>

  [Last accessed: 10/1/11]

- [Internet] European Association for Machine Translation

  Available from: <http://www.eamt.org/mt.php>

  [Last accessed: 10/1/11]

- [Internet] Centre for Next Generation Localisation

  Available from: <http://www.cngl.ie/index.html>

  [Last accessed: 10/1/11]

- [Internet] Computing Dictionary

  Available from: <http://www.computing-dictionary.com>

  [Last accessed: 10/1/11]

- **Machine Learning - Systems that Improve Their Performance** [Internet] AAAI

  Available from: <*http://www.aaai.org/AITopics/pmwiki/pmwiki.php/AITopics/MachineLearning*>

  [Last accessed: 10/1/11]

- Peter Flom **What is a statistical model?** [Internet] Helium

  Available from: <*http://www.helium.com/items/1992549-regression-models*>

  [Last accessed: 10/1/11]