

Project Plan

Group 52

February 2, 2017

0. Team information

Team number 52

Contact person: Alexander Marchand-Melsom

Contact email : alexamar@stud.ntnu.no

Contact phone (24/7) : 90714116

1. Problem

1.1. Opportunity

Little has been done in the last couple of years to incorporate automated technologies for educational endeavours. At the same time professors all over the world spend much of their available time doing repetitive tasks like answering the same question time and time again. At the same time, students asking questions that have already been answered or whose answer is readily available on the courses learning platforms, have to sometimes wait an inordinate amount of time for an answer.

By making an automated system that analyses questions already available on currently used platforms, we want to minimise the effort spent on these excessive task so that more time could be spent on preparing and lecturing, whilst significantly cutting the response time.

1.2. Stakeholders

Our stakeholders are mainly divided into two groups:

The first group consists of professors, teachers and other educational personnel who get questions from students.

The other group consists of (the great) Pekka Abrahamsson and our project manager Audun Liberg, whose requirements we have to satisfy. These requirements include planning the project and executing it following their guidelines.

Our end-users are the people asking questions on the forum, hence essentially students.


Steven Stevensen	
GOALS <ul style="list-style-type: none"> To work as little as viable To have good grades To be informed of everything 	AGE: 21 OCCUPATION: Student PERSONNALITY: Lazy IT SKILLS: Expert
FRUSTRATIONS <ul style="list-style-type: none"> Slow response time from professors Bad accessibility to information resources Bad grades 	
BIO <p>Steven is a student at NTNU, and enjoys playing videogames. As such, he tries to work as efficiently as possible so as to have more time to play on his computer. Which is why he hates everything that slows him down, such as slow professors and scattered or hidden information resources. At the same time, he wishes to have good grades, so staying informed is primordial.</p>	

Figure 1: End-user persona

1.3. Requirements

The software systems aim is to provide an automated responses system for the students. In order to achieve this, we have following functional requirements:

- Ability to answer questions, that is, be able to use an existing platform (i.e. Piazza) to interact with the end-users
- Ability to fetch relevant information to formulate an answer, through interaction with websites (i.e. Itslearning) and forum (i.e. Piazza)

- Ability to formulate a sensible and relevant answer based on gathered information
- Ability to store relevant information in a database

Based on this and the end-user persona, we may formulate several non-functional requirements:

- Usability: the use and adoption of the system should be as simple and as pleasing as possible
 - It should be possible to seamlessly implement the system into an existing framework, i.e. a form of plug-and-play
 - Whilst being the least bothersome for the teaching personnel, it should still be pleasing and simple to the students
- Correctness: the answers should be as correct and relevant as possible
 - The system should be smart enough to correctly match the questions to available knowledge (and thus formulate an adequate answer)
 - It should collect all relevant information correctly
 - The information should be kept up-to-date by routinely checking for new information
- Availability: the system must be much quicker to answer than a professor or assistant:
 - The system should frequently check for new questions
 - The system will need to be optimised so that retrieving relevant information from the database should take as little time as possible

Story ID	Story	Estimate	Priority
T1	As a user*, I want to know if my question has already been asked and answered or whose answer is on the course's pages without having to check myself, and get said answer	10	1
T2	As a user, I want to be able to ask a question on my favourite forum Piazza without having to install anything else	6	2
T3	As an administrator*, I want to be able to define which websites should be used as information source for the system	10	3
T4	As a user, I want the answers to be as correct as possible	7	4
T5	As a user, I want the possibility to point out that the system's answer did not satisfy me, prompting a professor to answer me	3	5
T6	As an administrator, I want to be able to endorse or invalidate the system's answer	1	6
T7	As a user, I want to have the quickest response time	5	7
T8	As an administrator, I want to have as little work as possible to do to implement the system	3	8

*: here user refers to the students, and administrator to the teaching personnel

Figure 2: Product backlog

2. Solution

2.1. Deliverables

Following deliverables are expected:

- The present project plan, particularly the product backlog
- Executable software system
- Poster
- Advertising video
- Final report

Differences in experience and rapidly-approaching deadline limit us to use technologies that we commonly have experience with or that we can learn easily. Such technologies include but is not limited to:

- API.AI (as a framework for creating the bot)
- Scrapy for webscraping the questions needed to compare
- mySQL to save data from the scraping

We may identify following technical constraints:

- The system shall be available nearly 100% of the time, 24/7
- The system will work on existing platforms and infrastructures
- The system will only use data available on the forum and the specified course websites
- The system will work with minimal maintenance

2.2. Work

Release Due date	Story ID or other task	Description Remarks	Estimated ressource use	Actual ressource use
27.04.17	Poster	Make the poster	4h	4h
	Project planning	Defining the project plan	10h	12h
	Interviewing	Interviewing professors.	1 h	1 h
	User stories	Writing user stories	6 h	
	Video	Shooting and editing the advertising video.	20 h	
	T4	Make the bot answer correctly using a given database	40 h	
	T3	Fetching data from websites to create a functional database	50h	
	T1 and T2	Make the bot interact with Piazza	60h	
	T5 and T6	Enabling contestation of the bot's answer	20h	
	T7	Optimisation	25h	
	Pre-testing	Pre-testing	10h	
	Post-pre-test report	Change system if needed	20h	
	Public beta	Pre-release and testing	1 week	
	Beta report	Report on test	10h	
	Meetings	Meetings to discuss changes to make to the plan	10h	
	Final package	Collect everything into a package	5h	
	Total		291h	

Figure 3: Activity plan

2.3. Team

Alexander Marchand-Melsom: Team leader, documentation responsible, developer

Daniel Sandberg: Back-end responsible, developer

Duong Bao Nguyen Mai: Front-end responsible, developer

Shivam Verma: Web-interaction responsible, developer

2.4. Way of working

We will use GitHub to coordinate work on the software itself (link: <https://github.com/alexamar0714/TRYBOT>), as well as Trello for organisation purposes (link: <https://trello.com/b/FoBU7o33/project>).

Communication plan

Seeing that we cannot work on this project every day, it seems unrealistic to have daily SCRUM meetings. Instead, we will communicate extensively using Facebook. There is going to be one weekly meeting where we will discuss what has been done, what obstacles have arisen, how much progress we have made, and what the plan is for the following week. In addition to this, there will be a monthly project control meeting to assess how the project as a whole is moving and eventually what changes need to be done to deliver on time. Meetings with our teaching assistant will be held every two weeks.

Quality assurance plan

In order to develop a product of good quality, we will enforce following procedures:

- Regular testing of working code
- System-wide testing when adding a new feature
- Public-beta test
- Follow coding conventions
- Regular meetings
- Clear communication

Risk	Means to prevent	Action and responsible
Failure to meet scheduled requirements	Controlled through weekly progress monitoring.	When schedule is not followed, the team should decide which stories or parts of the system should be dropped. Team leader is responsible to initiate the discussion.
Bugs when merging different parts of the program	Communication and use of adequate coding principles.	Developers must inform each other on what kind of output is to be expected from their program, and program in accordance to others' code.
Critical failure during public beta	Regular tests of the system as the project advances.	Developers are responsible to test their parts of the program individually and assure that there is no problem. Team leader must organise system-wide tests every time a new component is added.
Business misunderstanding	Clear communication.	Developers are required to analyse and break stories into tasks until they are understandable and programmable. Team leader sees that everyone fully understands the requirements and stories.
Technology	Lack of experience on used technology and tools prevented through pair programming and use of consultants when necessary.	Use of extreme programming method "spiking" to pre-test probable solutions and techniques. Developers are required to inform team leader about uncertainties, who is responsible to request consultancy when necessary. Team leader sees that time is allocated for training and reviews.

Figure 4: Activity plan