

---

# Project Planning

**Team: L02\_04**

Alley Dismay, Ka-Kit Jason Cheung, Pak Sun (Harrison) Fok, Zenan (Will) Ji, Zongyang (Eddie) Li

October 22, 2017

---



---

## Table of Contents

Product Backlog .....	2 - 15
Sprint Backlog .....	16 - 18

---

## Product Backlog (Version 0)

### Personas

#### Student Personas:

##### Sara Bai

- 20 year-old student, female, Canadian-born-Chinese, in 2nd year at UTSC.
- Speaks English fluently.
- Specialist in Sociology with a 2.5 CGPA.
- Loves to read long papers and write essays, but hates math.
- Low technological competence, only uses her computer to watch videos on YouTube, basic usage of MS Word to write essays and uses default web browser on computer to check UTSC Portal.
- Uses a very basic Nokia phone to make phone calls and text.
- She always prints her homework off the computer because she hates reading off a computer screen.
- Takes STAB22 as an elective, since she was told it was an easy A.
- Doesn't want to spend too much time learning software for an elective course, just to complete assignments.
- Wants to just use her uToronto credentials as opposed to making a new account as it would be too much work.
- Likes to divide her assignments and homework into several parts to complete them in smaller chunks, over time.
- Would rather do hardcopy assignments because she believes using software will complicate things.
- Likes to be prepared and caught up for every lecture and before doing every assignment.

---

## Lucas Liu

- 20 year-old student, male, from China, in 3rd year at UTSC.
- Speaks English and Chinese fluently.
- Double major in Statistics and Physical Sciences
- Introvert and autistic, never wants to communicate with people and enjoys staying alone in a room.
- Good computer skills: uses applications like the Microsoft Suite, Dropbox, Skype, computer games
- Typically too lazy to do homework because it's too boring and not engaging enough, would rather play video games like Dota 2
- Very interested in exploring new software and reading about new technologies.
- Works 25 hours a week at a retail store, trying hard to balance work and school.
- Likes to do a lot of practice and do as many problem sets as he can to prepare for tests and understand material
- Hates when professor give limited examples to work on
- Wants to be able to redo assignments to get the best possible mark.
- Appreciates how software looks and loves to use aesthetically pleasing software.
- Doesn't always have internet connection though, so prefers to have a desktop app rather than web app

---

## Alice Smith

- 22 year-old student, female, single Irish-Canadian.
- Lives on residence, with constant access to the school's wifi.
- Speaks English and Irish Gaelic.
- Specialist in Statistics, learns new things quickly and has strong multitasking abilities.
- Passionate about academics, outgoing and spends her free time getting her work done as soon as possible, tries to learn the fastest way of doing everything.
- Loves technology and has access to the cutting-edge of technology.
- Proficient in MS Suite, Photoshop, Final Cut Pro, and numerous other softwares.
- Only likes to do assignments once and disregards them afterwards, not looking at them again.
- Gets frustrated easily when it takes multiple steps to do something simple.
- Is a visual learner, so likes to look at graphs and charts which will help her learn the material better.
- Likes to work on mobile devices like her phone or tablet because its more convenient for her to carry around, rather than a desktop or laptop.
- Likes to keep track of her performance in assignments and other course work she has done.

---

## **Professor Personas:**

### **Professor Obi Wan-Too-Thrie**

- 65 year-old professor, female
- Speaks English and Hindi
- Works as a professor in the Statistics Department where she mainly teaches introductory statistics courses.
- Prefers to use simple software that is straight forward and uncomplicated.
- She can only perform tasks when someone teaches her and follows the exact same steps
- Little exposure to technology, no cell phone and uses very basic functions in MS Word and Excel to enter grades of her students.
- Teaching style: lets students practice on their own to figure out their own method of approaching problems.
- Doesn't use software that is not engaging or interesting aesthetically
- Currently has students print out exercises to do and hand in at tutorial, but those take really long to be marked (2-3 weeks)
- Teaches classes as large as 1000 students and is usually unable to go over exercises in lecture
- Wants to be able to generate many different questions and types for students through the software, so that students can work them out at home.
- Wants software to be intuitive and familiar to the current software she uses, like MS Word and Excel.
- Believes that gamification(virtual rewards) of exercises/ assignments will increase student's interest in learning and course content.

---

## Professor Bob

- 35 year old stats professor, male with no wife or kids
- works at a the university of Toronto with a degree in both Computer Science and Statistics as a Professor and researcher
- Prof Bob is a full time professor that teaches multiple first and second year courses on Statistics
- Likes to be very organized, likes predictability and things to be done quickly and efficiently
- Speaks English only
- Used to use software for assigning and having students answer questions, but it has become outdated, confusing to use and didn't have sufficient functionality (adding graphs, attachments to questions), which made him frustrated and stopped using it
- good computer skills in the Microsoft Suite, multiple IDEs and terminal
- Has at most 100 students in each class and only a couple of TAs to mark course work since Prof Bob is constantly busy doing research and writing papers
- wants to be able to quickly and easily create and upload questions and distribute them amongst all students
- wants students to be able to provide immediate feedback once students finish an assignment
- Likes to see how the class is doing and what material they are struggling with
- his teaching style is that he prefers students learn by example, rather than memorizing theorems, hoping using the system will allow for abundant question generation to provide many examples to learn and prepare with for exams
- want to give students as much practice / retries on different assignment questions until the deadline of the assignment

---

## User Stories

- As a Professor (Obi-Wan-Too-Thrie), I want to be able to create different types of questions like multiple choice, fill-in-the-blanks, matching, etc.
- As a Professor (Obi-Wan-Too-Thrie), I want each student's version of each question to be different, by having randomized numerical values.
- As a Professor (Bob), I want to assign graphics to questions.
- As a Professor (Bob), I want to be able to add/edit/delete questions from my database.
- As a Professor (Bob), I want to be able to select questions from the database and create an assignment (a collection of questions) for the students to answer.
- As a Professor (Bob), I want to view/edit/remove questions from my database after I've created them.
- As a Professor (Bob), I want to be able to store all my questions and the corresponding answers in a database by topic.
- As a Professor (Obi-Wan-Too-Thrie), I want to register a professor profile.
- As a student (Sara), I want to be able to make a student profile.
- As a Professor (Bob), I want to toggle the visibility of assignments for students, so that students won't see drafts of the assignments.
- As a Professor (Bob), I want to set a timeframe for when assignments can be completed. (i.e. If it's 1 week of visibility. At the end of 1 week, the assignment's visibility is automatically turned off).
- As a student (Alice), I want to be able to see a list of assignments that I have completed and ones that I have not yet completed.
- As a student (Sara), I want to know the deadline of each posted homework.
- As a student (Sara), I want to view which homework is graded.
- As a student (Lucas), I want to be able to answer the questions using this system and submit the assignment.
- As a student (Sara), I want the online homework system to allow me to save my current answer and quit an unfinished homework.
- As a Professor (Obi-Wan-Too-Thrie), I want a set of questions belonging to an assignment to be automatically marked as soon as a student completes it, so that they get immediate feedback.
- As a professor (Bob), I want to be able to store the student's highest, most recent grade on an assignment in the database.



- 
- As a student (Sara), I want to know the mark of all the tests and the solution of the last test done.
  - As a Professor (Bob), I want to see how many students get each question right or wrong, so that I can gauge the class performance.
  - As a Professor (Bob), I want assignment feedback to consist a list of correctly and incorrectly answered questions, the number of questions answered correctly in percentage and also the correct answers for questions that were answered incorrectly.
  - As a student (Lucas), I want to be able to redo assignments with new questions until the deadline so that I can get the most practice on a topic.
  - As a Professor (Obi-Wan-Too-Thrie), I want one occurrence of each selected question in a single assignment.
  - As a Professor (Obi-Wan-Too-Thrie), I want an assignment to re-generate random values each time a student attempts an assignment.
  - As a Professor (Bob), I want to be able to create multiple lectures for each class I teach, if I need to.
  - As a Professor (Obi-Wan-Too-Thrie), I want to be able to add students to a class, so that they can view and complete any assignment questions in the course.
  - As a student (Sara), I want to login to the online homework system using my UTORid and password.
  - As a Professor (Obi-Wan-Too-Thrie), I want to create badges and rewards for students depending on their performance on assignments, so that the gamification aspect increases interest.
  - As a student (Alice), I want to receive a notification/email indicating when a new assignment has been uploaded, so that I can finish it as soon as possible.
  - As a student (Sara), I want to know the related chapter of each posted homework.
  - As a student (Sara), I want to be able to save the assignment as a PDF, and other formats and print it so that I can work outside the app.
  - As a student (Lucas), I would like to return to where I stopped when I reopen the app.
  - As a student (Alice), I want to track and see my progress on an assignment or set of questions with a progress bar.
  - As a student (Sara), I want the online homework system to have a tutorial to teach me how to use this system when I first use it.

---

## Decomposed User Stories for the first sprint

### User Story 1: As a Professor (Bob), I want to register a professor profile. (5 story points)

#### Tasks:

- a) Design a form / window for registering profiles that takes in information for the profile. Possible options for fields are: **(1 Story point)**
  - i) Email
  - ii) First name / Last Name / Full Name
  - iii) Have a submit button for the user to process the action
- b) Implement Professor class to generate professor objects that store the information collected by the form and has methods to do things like. (helper functions to write and retrieve professor information to a database.) **(2 Story points)**
- c) Set up the form to, when the submit button is clicked, to take the information in the form and generate a corresponding Professor object. **(1 story point)**

#### **Dependency: a, b**

- i) implement the back-end methods to create a professor object with the filled in values of the form, and insert it to the database.
- d) Create a form / window to read the information of a professor object and display it. And likely allow editing. Perhaps repurpose the Profile Registration form. **(1 Story point)**

#### **Dependence: a, b, c**

---

## **User Story 2: As a student (Sara), I want to be able to make a student profile. (4 story points)**

### **Tasks:**

- a) Split the general profile form into two versions **(1 story point)**.
  - i) create a user interface similar to the professor one with basic fields of name, email and student number for students
  - ii) when a student is registering, show only the student specific fields required to make a student profile.
  - iii) use the Tkinter framework to develop the UI with the required fields and buttons.
- b) Create a option menu with Professor and Student profile choices so the user can select what kind of profile they're making. **(1 story point)**
- c) Create a super class that student and professor objects inherit from. **(1 story point)**
  - i) have an attribute that indicates whether or not the user is a professor or student.
  - ii) implement student and parent classes and allow them to inherit from the super class called User.
- d) Take the values provided in the fields and create a student object, and then insert it into the database. **(1 story point)**

### **Dependency: a, c**

- i) create database helper methods to insert the created student object into the database given its values.
- ii) Connect the submit button from the form to the back end methods to store the created user into the database.
- iii) would be calling the generic user class' insert into database method.

---

## **User Story 3: As a Professor (Bob), I want to be able to create different types of questions like multiple choice, fill-in-the-blanks, matching, etc. (8 story points)**

### **Tasks:**

- a) Design and implement the UI for letting the professor choose their type of question. **(1 story point)**
  - i) Draw wireframe/sketch the UI out on paper
  - ii) implement the UI with Tkinter framework
  - iii) Have a radio button for each type of question the professor could create.
- b) Create a generic question class that will hold the general attributes of a question. **(1 story point).**
  - i) each type of question will be a subclass that inherits from this class.
  - ii) this general question class has attributes like a boolean value for if the question was answered correctly, a field of the question and answer for the particular question.
  - iii) generic question class should have a method that inserts the created question into the database.
- c) Add the corresponding subclasses for multiple choice, fill-in-the-blanks and matching questions. **(1 story point)**
  - i) implement these classes with their question specific fields and have it inherit from the generic question class.
  - ii) for mc have an array to hold the options, match questions have a dictionary to hold the key:value, question:answer pairs, fill-in-the-blanks can use a box # : answer key:value pair as well.
- d) Design and implement the forms/UI for each different type of question (mc, fill-in-the-blanks and matching questions) for the professor to create a question with. **(3 story points)**
  - i) draw wireframe/sketch the UI out on paper
  - ii) implement the UI with Tkinter framework
  - iii) Do this for each type of question( ie. Multiple choice, fill-in-the-blanks, matching)
- e) connect the front end to the back end methods. **(2 story points).**  
**Dependency:** a, b, c

---

## **User Story 4: As a Professor (Bob), I want each student's version of each question to be different, by having randomized numerical values (5 story points)**

### **Tasks:**

- a) Implement a random function to randomize the values of a question each time it's being called. **(3 story points)**
  - i) implement a method to take the upper and lower bounds of a random number to be generated.
  - ii) make a call to this randomize value function each time a question is retrieved from the database that stores the questions.
- b) Return a question object that has the newly generated random values to be used by the professor. **(1 story point)**
  - i) Create a new question object that has these random values as a part of the object's attributes.
- c) Implement UI to allow the professor to choose a custom range of randomize values for a particular question and connect the back-end to front-end. **(1 story point)**
  - Dependency: a, b**
  - i) wireframe/sketch the UI with text fields to input a upper and lower bound.
  - ii) the menu should display the question that will have the randomized values. (use the database select helper methods)
  - iii) have a submit button for the professor to submit his option
  - iv) connect the front and back-end methods

---

## **User Story 5: As a Professor (Bob), I want to be able to select questions from the database and create an assignment (a collection of questions) for the students to answer. (8 story points)**

### **Tasks:**

a) create an assignment class that has attributes and methods for an assignment. **(1 story point)**

- i) have a field indicating the number of questions selected for the assignment.
- ii) have a list to hold the chosen questions in an assignment.
- iii) attribute to indicate deadline and start date

b) Implement method to create, remove questions, which will be the assignment. **(2 story point)**

#### **Dependency: a**

- i) implement a method for the assignment class use the database select helper to retrieve questions from the database and append it to a list that will hold the questions for an assignment.

c) create UI and implement method that displays all the questions in the database that the professor can choose from to add to an assignment. **(1 story point)**

#### **Dependency: a, b**

- i) wireframe a UI to display all questions that are present in the database.
- ii) create a method to select all from database and print to the screen.
- iii) link button to the method implemented in b)

d) Create UI to display an individual assignment with all of its questions. **(1 story point)**

#### **Dependency: a, b, c**

- i) implement method to print out all values in the array holding the question to an assignment.

e) Implement database methods for assignment class. **(3 story points)**

#### **Dependency: a**

- 
- i) create model/table in the database to hold the the attributes created in the assignment class.
  - ii) create insertion, retrieval methods for an assignment object into the database.

---

## Sprint Backlog

Team of 5 developers: Andy, Eddie, Harrison, Jason, Will:

- Andy is able to complete 7 story points a week
- Eddie - 6 story points a week
- Harrison - 6 story points a week
- Jason - 5 story points a week
- Will - 6 story points a week

Developers will work for 6 days a week and the length of the sprint is 1 week

### Team Velocity

The team velocity will be 30 story points a week.

### Provisional Schedule

Tasks	Day 1	Day 2	Day 3	Day 4	Day 5	Day 6
1a	Andy:1					
1b		Andy:2				
1c			Andy:1			
1d				Andy:1		
2a			Jason:1			
2b				Jason:1		
2c					Jason: 1	
2d						Jason: 1
3a	Eddie:1					



3b		Eddie:1				
3c			Eddie:1			
3d				Eddie:1		
3e					Eddie: 2	
4a		Will: 3				
4b			will: 1			
4c					will: 1	
5a	Harrison : 1					
5b	Harrison : 2					
5c		Harrison: 1				
5d			Harrison: 1			
5e			Harrison: 1	Harrison: 2		

Based on this provisional schedule, 4 user stories will be completed by the end of the sprint, with 28 user stories burned.

# Provisional burndown chart

Provisional Burndown Chart

