

# PISTACHIOS DOCUMENTATION

*Courtney Lum, Luke Ellis, Rebekah Chow*

## **1. USER STORIES**

### **1 User Story Point = 2 hours**

**Epic user story:** A university can obtain course feedback from students taking that course through a survey system.

**ID:** US1

**Name:** Login to the survey system

**User-story description:** As the administrator, I should have a specific login that provides me with administrator tools, so that only I have access to this and not students.

**Acceptance criteria:**

- The administrator should be provided with a login box that prompts them to enter their id and password
- If login is successful, the administrator should be provided with the tools they require
- If not then the page should give an error and let them try again, the user should be provided with at least 3 tries before it times out.

**Priority:** Essential

**Size:** 2 user story points

**ID:** US2

**Name:** Create the course questions

**User-story description:** As the administrator, I should be able to create courses and its required survey questions, so that I can receive feedback from the students

**Acceptance criteria:**

- The administrator should be able to add multiple-choice answers or fill in the space blanks for students to complete
- The administrator should be able to edit the questions or answer boxes if required

**Priority:** Essential

**Size:** 2 user story points

**ID:** US3

**Name:** Create a survey form

**User-story description:** As the administrator, I should be able to create a survey form so I can ask students questions and collect feedback

**Acceptance criteria:**

- The administrator should be able to create forms for any course running during a semester
- The administrator should be able to use some or all the questions from the generic question pool

**Priority:** Essential

**Size:** 2 user story points

**ID:** US4

**Name:** Make available a link to the created survey

**User-story description:** As the administrator, I should be able to create a link to the created survey so that any respondent can use the link to fill out the survey

**Acceptance criteria:**

- The administrator should be able to make the survey available to anyone from the public

**Priority:** Essential

**Size:** 1 user story point

**ID:** US5

**Name:** Completing the survey

**User-story description:** As a respondent, I should be able to complete the survey any number of times so that I can provide as much feedback as I want

**Acceptance criteria:**

- The respondent should be able to repeatedly complete the survey
- The respondent should be able to go back to the start of the survey at any time

**Priority:** Essential

**Size:** 3 user story points

**ID:** US6

**Name:** Reading responses

**User-story description:** As the administrator, I should be able to read responses from the survey so that I can collate the results

**Acceptance criteria:**

- All active responses should be accessibly
- Should be able to produce a statistical report of survey results
- Authentication required to access the results so the public are unable to view them without being provided with a link

**Priority:** Essential

**Size:** 2 user story points

**ID:** US7

**Name:** Availability of link to results

**User-story description:** As the administrator, I should be able to create a link to the survey results so that they can be made available to the public

**Acceptance criteria:**

- Link should be the only way that the public can access the results
- No other authentication is to be required for them

**Priority:** Desirable

**Size:** 1.5 user story points

**ID:** US8

**Name:** Create a bug report form

**User-story description:** As the administrator, I should let the user of the form report any problems they are having with the survey so that they can be passed on to the administrator to be fixed.

**Acceptance criteria:**

- The administrator should be able to save these bug reports
- The administrator should be able open and read these bug reports when it suits them

**Priority:** Future/Optional

**Size:** 1 user story point

## **2. ITERATION 1: CHOSEN USER STORIES**

User Story 2: Create the course questions

User Story 3: Create a survey form

User Story 4: Make a link available to the created survey

User Story 5: Completing the survey

### **3. ALLOCATION OF RESPONSIBILITIES**

Courtney and Luke – US2, US3, US4, US5 using OO Method

Rebekah – Documentation

### **4. STAND UP MEETINGS**

Week 6 Meeting: 23<sup>rd</sup> of August 2017

- Read through assignment specifications for iteration 1
- Decided which user stories to implement (US2-5)
- Created a rough plan of outline of website
- Roughly allocated tasks to each individual

Week 7 Meeting: 30<sup>th</sup> of August 2017

- Presented progress on allocated tasks
- Finalised tasks to be handed in on due date
- Finalise outline of website
- Discussed design and visual elements for website

### **5. ITERATION 1: LOG OF TASKS**

23/08/17 – Created hand drawn drafts of diagrams of class diagram, use case diagram and sequence diagram. To create class diagram, first determined which classes would be used and the corresponding attributes and methods, then worked out relationships between classes. For use case diagram, defined actors and processes and then created links between the two. For sequence diagram, wrote out steps which admin and respondent would follow, then wrote them out following scaffold of a sequence diagram.

24/08/17 – Initialized the index page with two buttons – ‘Admin’ and ‘Student’, which respectively link to the admin and student page. To redirect, had to use javascript code – location.href so that when the respective button is clicked it links to the correct page.

Then worked on the admin login page, by adding two input boxes that accept a username and password, these would be verified using string comparison. If it passes, then the admin would be redirected to the admin tools page.

26/08/17 – Finalised diagrams, drawing them on a tablet and sending them to other group members to ensure nothing has been forgotten and that it corresponds with the code being written.

29/08/17 – Added security measures to the admin tools page. As its possible to land on the admin tools page by just typing the correct link, so added an extra feature that checks that the user is logged in as ‘Admin’ before they can access that page. If they aren’t then it would redirect the user to the login page and display an error as to why they can’t access that page. To do this we used the sessions feature in flask to check that they were logged in.

1/09/17 – Created the page where the admin can create questions, to do this the admin just enters the question they want then the server checks that the question has not already been added, if it has then an error pops up, else the question would be added to the json file and a ‘success’ message would display. Also added the ability for the admin to check the questions that they have added, this is done by the server accessing the respective json file and then displaying the results in that file on the screen.

Then added the page for the admin to create the survey form, first needed to convert the courses.csv file into a list that we can read from, then we transpose this list into a drop down menu in the html template. Next the questions from the json file would be extracted and passed into a drop down menu in the html template. From there the admin picks the course and the question they want to add, if it that course already has that question then error message is displayed, else the

question is saved as a key to that course value, i.e. a dictionary type. Then a link to that course survey is generated so that the admin can share it to the students.

For the students to fill out the survey, the server takes that dictionary and extracts the respective question values to that course key, then adds the multiple choice answers to those questions as radio buttons.

2/09/17 – Added a property decorator that checks that the session is that of an admin type, this saves us from repeating code on each page that needs admin qualification.

5/09/17 – Submitted all documentation onto the git for final checks from the group before hand in tomorrow.

## **6. ITERATION 2: CHOSEN USER STORIES**

**ID:** US1

**Name:** Login credentials

**User-story description:** As a user with access rights, I should be able to login in and then be redirected to the specific page depending on my role, so that I can access the webpage.

**Acceptance criteria:**

- When accessing the survey page I should be redirected to the login page if I am not yet logged in
- The user should be prompted with the ability to enter their username and password
- When entering the password, the output that I enter should be hidden so no one can see
- If login is successful the database should check the role of the user and redirect them accordingly to the right webpages
- If login is unsuccessful the page should give an error and let them try again.

**Priority:** Essential

**ID:** US2

**Name:** Seeing which courses are open

**User-story description:** As the admin I should be able to see a list of courses that are currently opened for answers so that I can find the surveys that need to be closed

**Acceptance criteria:**

- The correct courses should be displayed in a list format so that it can be easily read

**Priority:** Desirable

**ID:** US3

**Name:** Create questions

**User-story description:** As the admin I should be able to create a question so that I can add it to the pool of questions that are to be picked for making course surveys

**Acceptance criteria:**

- I should be able to make the responses to these questions as multiple choice or text-based
- The questions should be unique, any duplicates should not be created

**Priority:** Essential

**ID:** US4

**Name:** Delete questions

**User-story description:** As the admin I should be able to delete a question so that I can modify the list of questions in the question pool.

**Acceptance criteria:**

- When the question is removed it should also be removed from the course surveys that use that question

**Priority:** Essential

**ID:** US5

**Name:** Create the course survey

**User-story description:** As the admin I should be able to create a survey for a course so that the course can receive feedback from the students

**Acceptance criteria:**

- The admin should be able to add a question to the course
- The admin should be able to delete a question from the course
- A course cannot have duplicate questions

**Priority:** Essential

**ID:** US6

**Name:** Changing status of a course survey

**User-story description:** As the admin I should be able to change the current status of a course survey so I can manage who can view the survey

**Acceptance criteria:**

- Admin should be able to open a course survey so that students can answer
- Admin should be able to close a course survey so students cannot view it anymore
- Admin should be able to put a course survey for review so that the staff can check before it is published

**Priority:** Essential

**ID:** US7

**Name:** Seeing which courses to review

**User-story description:** As a staff I should be able to see which course surveys I need to review so that I can check that it is okay before publishing it for student answers

**Acceptance criteria:**

- Staff should be able to see the list of course surveys they need to review
- When the course survey has passed the review, it should disappear from the list of surveys the staff needs to review

**Priority:** Essential

**ID:** US8

**Name:** Students current courses

**User-story description:** As a student I should be able to see the list of courses I am undertaken this semester so that I can answer the surveys related to it

**Acceptance criteria:**

- Students should only see the courses they are in
- If the student tries to access the survey belonging to a course they are not part of the webpage should prompt them with an error

**Priority:** Essential

**ID:** US9

**Name:** Answering survey

**User-story description:** As a student I should be able to answer a survey so that I can provide feedback for the course

**Acceptance criteria:**

- Only students part of that course should be able to answer the survey
- If the student has already answered the survey they should be prompted with an error message that prevents them from answering again
- If a staff or admin tries to answer the survey they should be prompted with an error and prevented from answering it

**Priority:** Essential

**ID:** US10

**Name:** Logging out

**User-story description:** As a user I should be able to log out from the system so that no one else can use my session

**Acceptance criteria:**

- Only users logged in should be able to log out
- Once the user has been logged out they should be redirected to the login page

**Priority:** Essential

## **7. ITERATION 2: LOG OF TASKS**

14/09/17 – Using sqlite, created a table that maintains the list of questions and another table that tracks the multiple choice answers associated with the question, these were linked using the questionID as the foreign key. Then created a file that stored the commands related to the Model View Control Architecture, to this the ability to add questions, delete questions and add answers were created.

Planned out all documentation that is required for the next deliverable. Created hand drawn versions of sequence diagrams and class diagrams. Sequence diagram included ones for creating questions, surveys, reviewing surveys, completing surveys and closing surveys. Class diagram was similar to the previous iteration however, had features added as requirements changed.

16/09/17 – Added the ability to add questions to the course surveys from the pool of questions, as well as the ability to remove these questions from the course survey and to be able to delete questions from the question pool.

17/09/17 – Changed the design of the login page to make it more visually pleasing, also modified the way in which passwords are entered so that the input to the screen is hidden. Made a new table that contains the login details from the passwords.csv, i.e. username, password and role, so as to authenticate any logins.

19/09/17 – Used flask-login to manage the accessibility rights of users to certain webpages, i.e. when a user successfully logs in they should be redirected to the webpage that corresponds to their roles. Also a user should not be able to enter webpages that are restricted to them, this is done by using a function wrapper that checks the role of the current user to determine if their role matches the one of the webpage.

21/09/17 – Updated the student dashboard by presenting the student with the courses they are currently enrolled in for the semester, they are also given the ability to answer the surveys related to that course.

22/09/17 – Created a new database table that relates to a course survey's current availability, i.e. whether it is in review, open or closed. Then added a toggle switch that allows for the course survey to be opened, closed or sent to be reviewed.

23/09/17 – Transferred all documentation onto computer to be made neater. Shared documentation with group members for final checks to ensure everything was added.

24/09/17 – Added the functionality that allows a student to answer a course survey once only. In addition to this, the system was modified so that only students in that course can answer that survey, hence preventing staff, admins and students outside of the course from answering it.

3/10/17 – Provided ability for creating multiple choice questions and short answer questions

6/10/17 – Created two more users to the system, the staff and admin, then provided these roles with the tools pertaining to their roles, i.e. staff can only review courses that need to be reviewed, and admin can add questions, add questions to course survey and push a course survey for review, open or close.

8/10/17 – Pushed all documentation to the git to ensure that everything was located in one place for the hand in date on Wednesday.