**Team Chundercats**

***ABTAHI DEBOER GOLUBOW PATEL RACHMAT STILL***

**User Stories**

**TEMPLATE - DO NOT EDIT - JUST COPY PASTE - OTHERWISE I WILL MURDER YOU**

**Title**

1. Description
2. Assumptions:
3. Tasks
4. Tests
5. Assignees
6. Estimation
7. Priority
8. Definition of Done
9. Baseline Story point estimation

All assignees for all stories are directly linked to their corresponding task number

-ex: assignee a) Bob -> task a) kick butt

-Bob’s job is to kick butt, and only Bob should be kicking butt

**#1 - PN:10 - Andre - (Access Testimonials)**

As a general user I can access testimonials in order to see what others think about the product and leave a testimonial myself

1. Description:

A Guest user can rate the software on a 5 star rating system. The extent a user can provide a detailed testimonial depends on the user type. (Review tasks) An immediate response will pop-up to the guest user as a confirmation that

the testimonial was published.

1. Assumptions:
   1. The team understands qt
   2. A baseline project is established in qt, and pushed to the master branch on git
2. Tasks
   1. Create a simple main menu GUI in qt
   2. Provide access to a screen that would display the testimonials
   3. Create a table to store the testimonials in our db
   4. Write queries to fetch the testimonials
   5. Write a function to populate our .ui with them
   6. Provide an interface to add a testimonial
   7. Write statements to store the new testimonials in our db
3. Tests
   1. Verify that all testimonials are being fetched from the db and displayed
   2. Verify that new testimonials are saved into the database after entry
   3. Verify that testimonials cannot receive rogue input
4. Assignees
   1. Andre
   2. Anthony
   3. David & Alex
   4. David & Alex
   5. Jacob
   6. Krupesh
   7. David and Alex
5. Estimation
   1. 20 hours
6. Priority
   1. LOW 10/10
7. Definition of Done

A simple executable in qt that can provide a main menu with an option to an interface for user testimonials. The program will then allow the user to view all testimonials, add a testimonial, and/or return to the main menu. The testimonials will be loaded from and saved into a db. The testimonial table with appropriate queries will be written and functional with our qt program. Test plan will be updated to include tests for this story and those tests will be executed until program is debugged

1. Baseline Story point estimation:
   1. Baseline story point estimate value of one = 2 hrs of development time for a single developer

**#2 - PN:9 - Andre - (Contact Us)**

As a guest I need to contact the administrator so that I can receive assistance when I need it

1. Description

The program need to provide access from the main menu to access a “contact us” screen that shows a team logo, team name, and contact information for the administrator. An optional FAQ section may also be implemented if development time permits. FAQ section would provide assistance for any user that is confused with the program

1. Assumptions:
   1. Basic main menu GUI is complete for our project
   2. Team logo is decided upon and is saved to our project as a png file
   3. Contact information is for our administrator’s email address
2. Tasks
   1. Create a ui to display a screen for the “contact us” page
   2. Get team logo png
   3. Attach and display the logo to the ui
   4. Edit UI window to display the team name and contact info
   5. Create an option to return to the main menu
   6. If time permits, create a FAQ page ui with pertinent information about the program
   7. As a premium user, I want access to contact the company in charge of the software directly. As a premium user, I also need a FAQs section that tailors to my needs and wants as a premium user, such as account settings, paid membership settings.
3. Tests
   1. Verify that the logo and information do not overlap when resizing the window
   2. Verify that the user can return to and from the main menu
   3. Verify that the team logo and info is displayed consistently between executions
   4. Verify that an admin user's contact info including logo, etc are able to be published.
   5. Verify that a guest user can access the contact portion of the software.
   6. Verify that a guest user can access the guest FAQs section for any help needed
   7. Verify that a non premium user can access the contact portion of the software.
   8. Verify that a non premium user can access the non premium user FAQs section for any help needed
   9. Verify that a non premium user's account information can automatically sync when they request for help or need to contact the company to cut friction with user.
   10. Verify that a premium user can access the contact portion of the software.
   11. Verify that a premium user can access the premium FAQs section for any help needed
   12. Verify that a premium user's account information can automatically sync when they request for help or need to contact the company to cut friction with user.
4. Assignees
   1. Andre
   2. Krupesh
   3. Krupesh
   4. Anthony
   5. Anthony & Andre
5. Estimation
   1. 10 hours
6. Priority
   1. LOW 9/10
7. Definition of Done
   1. N/A
8. Baseline Story point estimation
   1. Baseline story point estimate value of one = 1 hrs of development time for a single developer

**#3 - PN:2 - Jacob - (Main View Window)**

As a user I want the ability to see the shapes I create in a window. The shapes will have an ID and the window will be 1000 by 500 pixels.

1. Description

A user needs the ability to see the shapes they create. This will be done in a window sized 1000 x 500 pixels. The shapes will also have an ID above them. This ID will have the ability to be hidden or shown on a user’s demand. The coordinate system is defined such that the top left corner of the rendering area is located at point (0,0), the bottom right corner at point (1000,500). This will be done in a separate screen stemming from the main menu, and will load all the available shapes from the database.

1. Assumptions:
   1. The main menu UI is created and functional
   2. The database for the shapes is created
   3. All shape class files (including their derived types) are created (.cpp/.h pairs) and integrated into the qt project folder
   4. All shape classes are tested and compilable
   5. Our templated vector container for storing the Shape objects is created and tested
   6. The rendering window will be run based off of the qt\_basic\_drawing\_example.pdf example document
   7. The window will be separate from the main window
   8. Exiting the rendering window will not terminate the program
   9. Only 1 rendering window will be able to be opened at one time
2. Tasks
   1. Create a viewing window that is 1000 x 500 pixels inside of program window
   2. Create shape classes .cpp/.h
      1. Base class Shape
      2. Line
      3. Polyline
      4. Polygon
      5. Rectangle
      6. Ellipse
      7. Text
   3. Link the classes with the QPainter objects
   4. Update the viewing window every time a new shape is drawn / change is made
   5. Make text appear above shape telling you what the ID is
   6. Make button that toggles hiding and showing the shapes’ ID
3. Tests
   1. See if window shows up
   2. Can you “draw” shapes on window?
   3. Does window update on every new change?
   4. Are IDs shown above every shape?
   5. Do IDs move/delete with the shape?
   6. Does the button hide the IDs?
   7. Does the button show the hidden IDs?
   8. Repeat #d - g multiple times in different orders to check for glitches
4. Assignees
   1. Andre
   2. Anthony
   3. Alex
   4. David
   5. Jacob
   6. Krupesh
5. Estimation
   1. 20 - 25 hours
6. Priority
   1. HIGH 2/10
7. Definition of Done

Test plan is passed and the user can access the rendering window, see all graphical objects that were stored in the db with their corresponding IDs in the correctly sized window (1000x500). The user can consistently open and close the window from the main menu without glitches.

1. Baseline Story point estimation
   1. Baseline story point estimate value of 50 = roughly 1-3 hrs of development time for a single developer

**#4 - PN:1 - Jacob - (Load Shapes from File)**

As a user I want the program to display shapes based on a file

1. Description

The program should be able to display shapes based on a save file (or database). This file will be where all the relevant information is stored on the current display window. such as number of shapes, shape geometric dimensions, location, ID, etc. This file will be stored on a database server along with other shape files. The user will have the ability to choose a file to load or create a new file.

1. Tasks
   1. Create a main menu screen with:
      1. Welcome message
      2. “New Shapefile” button
         1. Goes straight to new file editing window
      3. “Load Shapefile” button
         1. Opens window to select file to open
      4. “Save Shapefile” button
         1. Saves file - See storyboard #10
      5. “Help” button
         1. “Tutorial” button
            1. Brings up a tutorial window
         2. “About” button
            1. Short description of program
         3. “Contact” button
            1. Opens contact window
         4. “Reviews” button
            1. Shows reviews of the program
   2. Create ability to load this file from database server.
   3. Write an input file loading function(parser) to transfer the data from the file directly to the database
   4. Write a query to load our vector of shapes from the remote database server
   5. Write a method to link the queries to the interface in qt
2. Tests
   1. See if the input file was loaded into the db correctly
   2. Check to see if the program can connect to the remote db consistently
   3. Check persistency of changes to database
   4. Check how program handles rogue input file data
3. Assignees
   1. Andre
   2. Alex & David
   3. Jacob,Alex,David
   4. Alex & David
   5. Alex & David
4. Estimation
   1. 25 hours
5. Priority
   1. HIGH 1/10
6. Definition of Done

The program can consistently load the shape data from the file and save to the remote database. Changes to the data in the program (admin) are persistent between executions and all further executions of the program load their data from the database. Passes test plan and test plan is updated.

1. Baseline Story point estimation

**#5 - PN:4 - David - (Move Shapes)**

As an administrator, I want to move the shapes and text in my rendering area using a move shape form

1. Description

The program needs to allow the administrator to login, open the shape rendering window, and only if logged in as administrator, move the shapes within the rendering area.

1. Assumptions:
   1. Stories 1 & 3 must be complete
   2. The vector class may need to be complete and functional
   3. The db management system will be decided by Alex
   4. The load from the file will not populate the vector but just the db, and then the db will populate the vector
2. Tasks
   1. Establish a remote db using a db management tool
   2. Create a tables for storing the Shape data
   3. Write queries for fetching the data
   4. Write a parser for the input file given in the project
   5. Connect the parser to the db
   6. Connect the load queries to the pt program’s vector
3. Tests
   1. Verify that the input file is accurately loaded into the dbs
   2. Verify that the db consistently take in the data accurately
   3. Verify that the user can load the data from db
   4. Verify that the admin can move the shape on the rendering area
   5. Verify that the shapes cannot be moved outside of the rendering area
4. Assignees
   1. Alex
   2. Alex
   3. Alex & David
   4. Krupesh & Anthony
   5. Jacob
   6. Andre
5. Estimation - 25 hrs
6. Priority
   1. MED - HIGH 4/10
7. Definition of Done

The program can upon start, load the shape data from the file into a vector, and the user can save the data into either a remote or local database. The administrator, once logged in, can open the window for the rendering area and the all stored shaped are displayed. The admin can then move the shapes around the rendering area

1. Baseline Story point estimation - 5

**#6 - PN:3 - Alex- (Add and Remove Shapes)**

As an administrator I want to Add or remove a shape from the rendering area.

1. Description

-Administrator clicks edit rendering area button (perhaps just called “Edit”)

-add/remove shape form pops up

-Administrator chooses whether or not they would like to add or remove a shape

-If remove is chosen, they will be asked to select which shape they would like to remove.

-If add is chosen, Administrator will be prompted on whether or not they would like to create and add a new shape, or add a previously existing shape.

-Form will contain input fields for all dimensions and properties of shape/text if Administrator chooses to create new.

-Necessary changes made in database

1. Tasks
   1. Create buttons and ui widget component that contains the add/remove form
   2. create list of currently rendered shapes/text with checkbox for remove
   3. create list of currently rendered shapes/text with checkbox for add
   4. Create form with fields for all relevant properties for shapes/text
   5. Ensure resulting changes are reflected in the rendering area
   6. Ensure any necessary changes are made to database
2. Tests
   1. Verify that add/remove form is always opened when edit button is clicked
   2. Verify that list of rendered shapes/text corresponds to what is being rendered
   3. Validate **ALL** input fields against input failure and malicious user input
   4. Verify that changes reflect user actions
3. Assignees
   1. Andre
   2. Anthony (micro version of shape report) (b and c are essentially same task)
   3. Anthony (micro version of shape report) (b and c are essentially same task)
   4. Alex
   5. Alex
   6. Alex & David
4. Estimation - 15 hours
5. Priority
   1. MED - HIGH 3/10
6. Definition of Done

Administrator can add, create and add, or delete shapes from rendering area in real time

1. Baseline Story point estimation

**#7 - PN:8 - Anthony - (Create Shape Report)**

1. Description

* New window pops up that outputs all shapes according to their id.
  + each newly created shape will receive an id in the order it was created. Ex: circle (id: 1) triangle (id: 2)
* The new window will pop up with different columns that show shape and id.
* Sequel will automatically sort the shapes for us.

1. Assumptions:
   1. Database for shapes will be done by Alex.
   2. Adding, deleting, and editing shapes and sending them to database will have to be complete.
   3. Save function that executes after adding, deleting, and editing shapes will have to be complete (User Story #10).
2. Tasks
3. Create button on main window that says “Print Shapes by ID”
4. Create a new window that pops up after user presses button from task A.
5. Make queries within sql that fetch shapes by id.
6. Create print function that outputs the shape in database on popped up window in task B.
7. Tests
   1. Make sure sql sorts the shape by id.
   2. Verify that the “Print Shapes by ID” button is shown on the Qt window.
   3. Verify that the print function works by seeing if it outputs the shapes in the database.
   4. Verify that a new window pops up with sorted shapes by id.
8. Assignees
9. Anthony
10. Anthony
11. Krupesh
12. Krupesh
13. Estimation

* 15 hours

1. Priority

* LOW 8/10

1. Definition of Done

* This will be done when the user is able to see an option that says “Print Shape ID” on the window and when a new window pops up that shows the printed Shapes by id.

1. Baseline Story point estimation

* Baseline story point estimate value of one = 1 hrs of development time

for a single developer.

**#8 - PN:6 - Krupesh - (List Shapes Created)**

No. 8 and No. 9 have been combined, see No. 9 below!

**#9 - PN:7 - Krupesh - (List Shapes Created)**

As a user I want to able to see the list of shapes that I created. I would like to see two seperate listing reports: one that lists the shapes by area, and one that lists the shapes by perimeter.

1. Description

* When using our product, the user should be able to generate a list of the shapes they created during any previous executions and/or the current execution.
* The shape listing report should list some characteristics the makes up the shape. (i.e. shape area/perimeter, shape type, shape id should be included etc.)
* For the list that gives priority to the shape area, list only the shape area along with shape ID and shape type; for the list that gives priority to shape perimeter, list only the shape perimeter along with shape ID and shape type.
* This is necessary so that when the user creates multiple shapes, they can keep track of what shapes they have already created.

2. Tasks

1. Create a class that dynamically saves the list of shapes by area or by perimeter.
2. An object of this class would need to be a some sort of an array/list datatype that is sorted by area or perimeter.
3. Save this list so it can pulled for multiple executions.
4. Needs to be dynamic so the user can add/delete to the list every time they wish to create/delete shapes.

3. Tests

1. Verify that the list is created with no entries in the beginning of the first execution.
2. Verify that both lists are created simultaneously when the user adds shapes.
3. Verify that the lists grows with single entries and/or multiple entries of shape.
4. Verify that lists get saved after every execution.

4. Assignees

1. Krupesh - create listing/report class
2. Andre - assist in class creation
3. Anthony - assist in class creation
4. Jacob/Alex/David - test the class with dummy shapes.

5. Estimation

* 15 hours

6. Priority

* MEDIUM 6/10

7. Definition of Done

* The list report updates according to how the user creates/deletes shapes.

8. Baseline Story point estimation

* What does this even mean?

**#10 - PN:5 - Anthony - (Save Changes)**

As a <user> I <need> <to be able to save all changes between executions>

1. Description
   1. When adding, deleting, and editing shapes, the developers need to make a save features that saves the changes made by the user. This is so the user will be able to continue where he/she left off when using the program.
   2. Save function will be done implicitly after every action. Do not need a save button.
   3. Both admin and guest users will be getting access to this save feature.
2. Assumptions:
   1. Create database using sequel that will have a different table for shapes and for testimonials. This will be done by Alex.
   2. Adding, deleting, and editing shapes feature should be complete.
   3. Adding, deleting, and editing testimonials feature should be complete.
3. Tasks
   1. Create save function that will be called after every action that involves shapes.
      * 1. Adding
        2. Deleting
        3. Editing
   2. Create save function that will be called after every action that involves testimonials.
      * 1. Adding
        2. Deleting
        3. Editing
4. Tests
   1. Verify that the database is accurate after adding, deleting, and editing shapes.
   2. Verify that the database is accurate after adding, deleting, and editing testimonials.
   3. Verify that an admin user can access saved data.
   4. Verify that a guest user can access saved data.
5. Assignees
   1. Krupesh
   2. Andre
6. Estimation
   1. 20 hours
7. Priority
   1. MEDIUM 5/10
8. Definition of Done
   1. This user story will be done when the save function executes after every action that involves adding, deleting, and editing testimonials. Database should accurately represent the changes made by the admin/guest user. This will be done implicitly without a save button.
9. Baseline Story point estimation
   1. Baseline story point estimate value of one = 2 hrs of development time
      1. for a single developer.