Final Report

CharityQ

Designed and Created by:

Capstone-004 Class of Spring 2018

For:

tagg

TOGETHER A GREATER GOOD

Contents

Index of Figures

Project Acknowledgements

The Capstone Class Spring 2018 Team would like to express appreciation to those who provided support as we worked on this project.

Dr. Sachin Pawaskar, Ph.D., MBA – Your expertise and coaching is what drives us towards success. We appreciate your help getting us fired up for this project. We appreciate everything you have done to ensure the project went on smoothly and the experience we gained through this process is valuable.

Fall 2017 Capstone team - We thank you for giving us a framework of the website and building the initial databases. That helped us sail through the remaining part of the project and add/tweak features to the website and produce the deliverables. Also we thank you for staying in touch with us to answer our questions regarding this project that we are assigned to take up.

Praveen Kumar – We thank you for your support to the Fall 2017 which helped us to enhance the application further.

Astrill Quadros– We thank you for your immense help ensuring the project stays on the right track throughout. You walked us through the process of learning new technologies, the Agile methodology and made sure we were accountable for delivering a quality product at the end.

Leslie and the entire team at TAGG – We thank Leslie, Holly, and the entire team at Tagg for being supportive and flexible as we worked through this learning experience together.

College of Information Science and Technology at University of Nebraska, Omaha (UNO) – What we have learned from the university will come with us as we move forward in our careers. Thank you for always having an open room for discussion. Thank you for the support and mentorship of all the teachers that brought us to this point.

Sponsors – Thank you for your generous donations which allowed us to professionally showcase our work.

Family and Friends – To the family and friends who have supported us through this, and previous semesters, thank you for your love and understanding. Our success is your success.

Statement of Work (SOW)

The Team along with the clients have developed a list of required features prior to beginning the project. Several meetings were conducted, and the client corresponded with the client liaison about their requirements via email and client meetings. A statement of work was prepared to determine the following aspects of the project:

- Scope of the Project
- Project Schedule
- Client Responsibilities
- Team Responsibilities
- Approval Process

- Communication Plan

Capstone Team of Class Spring 2018

Scrum Master	Bidisha Biswas
Project Manager	Mandeep Singh
Client Liaison	Abjijit Mohanty
Development Team	Ajay Khampariya
	Pooja Yalala
Testing team	Mounika Sreeperumbuduru
	Ajay Kanagala
Documentation Lead	Manmohan Rathore

The Client – TAGG

Our clients are the founders of Tagg and the ones have come up with the innovative thought of Charity Q named,

- Leslie Fischer
- Holly Baker

Problem Description

Charities and other organizations solicit businesses to make donations. Business owners can support these charities as a way of generating goodwill in the community. Also it helps them to receive federal tax exemption under section 501(c)(3) of Title 26 of the United States Code. The current process of deciding which charities to donate to and which to reject is decentralized, disorganized and manual. The following types of issues can occur:

- Businesses are unaware of the needs of the society
- Non-profit organizations are unable to get noticed
- Lack of communication mediums between the donors are requestors
- Requests can get lost or be overlooked due to inadequate storing systems
- A business owner may forget to follow up with an organization
- Requests received may have incomplete information

- Business owners may be overwhelmed by the number of requests received

Once the project has been undertaken by the Fall 2017 Capstone team, there has been improvements. A website has been built to enable requestors request for an amount from the business. The business was able approve or reject the request through the website. However, there are major bugs and shortcomings. Some of them were as follows:

- Businesses could not change payment methods
- Unable to create donation preference filters to enable auto-accept or auto-reject
- Set donation budgets
- Unattractive UI
- Incorrect information on dashboard

The project was transferred to the next 2018 Capstone team to incorporate major changes and fix the bugs so that it can be passed into production.

Current process

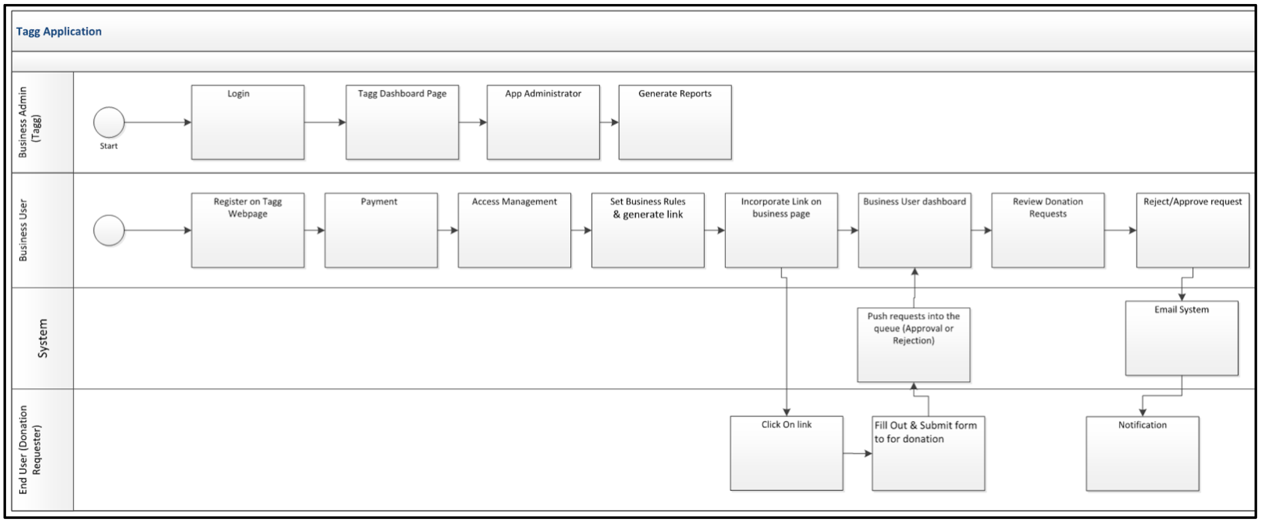


Figure 1: Representation of the process flows before the project starts

Proposed Process

For business:

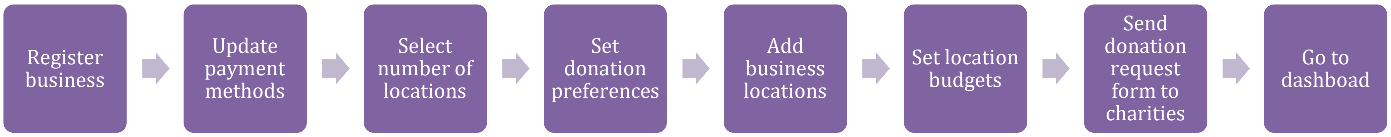


Figure 2: Business process

Once a business receives a donation request,

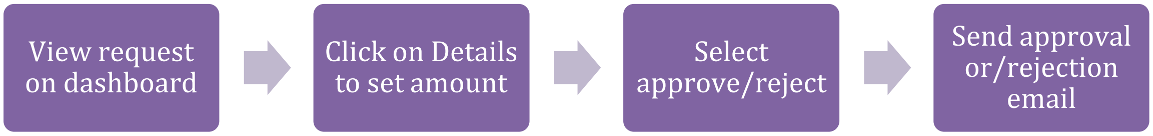

View request on dashboard
Click on Details to set amount
Select approve/reject
Send approval or/rejection email

Figure 3: Process to approve or reject donations

Keeping the process same as above we have added a plethora of new features that will make the website more convenient for the users such that it becomes a one-stop solution for the donation-management system.

We have uniquely identified three categories of users namely, CharityQ users (includes admins), Business users (includes admins) and organization users. The new features are as follows:

- Functionality for Stakeholder – CharityQ
    - Separate access privileges for CharityQ admins and CharityQ users
    - Admins can add and delete users
    - Users can add or remove email templates
    - Ability to add a coupon code to enable free trial.
    - Users can see the subscription status of registered businesses
- Functionality for Stakeholder – Business
    - Ability to delete users for Business admins
    - Business admins will be able to add users
    - Business admins can add multiple users and multiple business locations
    - Business users can send customized templates from the dashboard
    - The dashboard shows the count of
    - Businesses can set filters (such as budget) to approve or reject donations
    - Despite the donation request amount, businesses can choose a amount of their choice to donate
    - Businesses can opt for a free trial before paying subscription charges
    - Businesses can change their payment methods
- Functionality for Stakeholder – Organizations
    - Add a message on the screen that confirms that the donation request has been submitted.

Other changes:

- New logo and welcome page for CharityQ
- Revamped UI

Database model

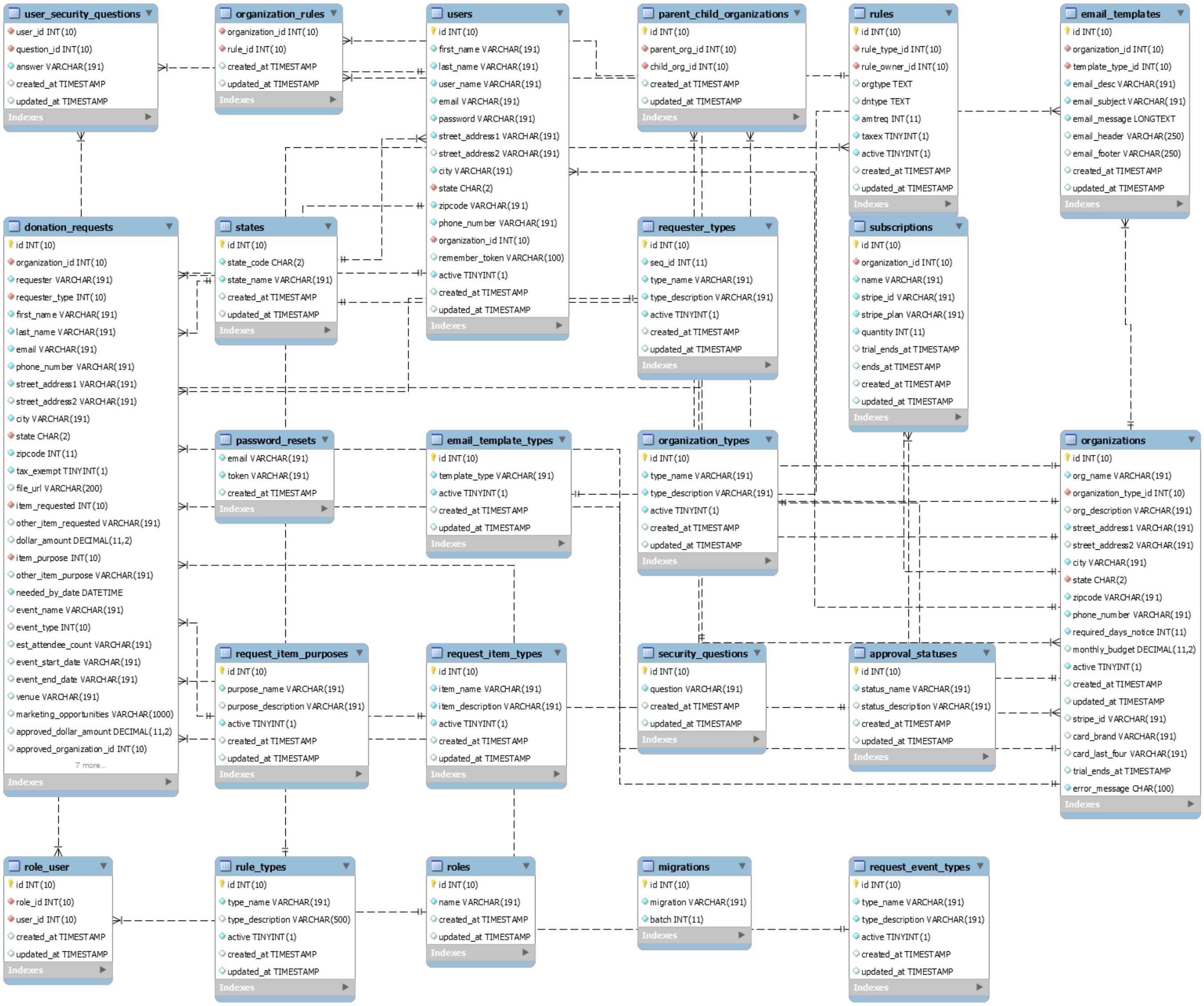


Figure 4: Database model

Scope of Project

The Statement of Work talks about the scope of the project. The features mentioned under proposed process are the ones that are successfully implemented in the project. The nice-to-have feature were to let organizations select business locations. Our team is able to deliver that as well.

Project Development with Scrum

One of the development methods is Scrum for agile management which is flexible, iterative and focused. The SCRUM Process is an iterative process developed as part of the agile methodology to accelerate the ability of teams to respond quickly to change (Pries, 2010). This process was used by the team to plan tasks and engage the client to assist in the process. The process considers the learning by the project team and client as they work on the project tasks.

Roles

An agile SCRUM process is generally composed of the following roles:

Client Liaison

- Liaison between the team members and the stakeholders
- One responsible for the correct creation of the product
- Communicate on daily basis with stakeholders, financiers and the team

Scrum Master

- In charge of observing team and guide them to follow the rules and methods of scrum
- Is always available for the team as an advisor
- Resolving Impediments and responsible for making sure that the final prototype matches client requirements
- Facilitate Scrum Ceremonies

Development Lead

- Responsible for making sure the technical team is on track
- Check code for errors
- Help the team during roadblocks
- Maintain Git Repository

Quality Assurance Lead

- Test the application thoroughly
- Generate error reports
- Ensure the application is bug-free before handing over to clients or before it goes to production

Sprint Cycles

The entire application is divided into various sprints based on the total length of the project. Each sprint is called a sprint cycle as the sprint process is iterative. Each sprint cycle is iterated to be two to four weeks long based on the requirements and overall length of the release. At the end of each sprint it is made sure that the features developed are eligible to be delivered to the client for further testing and feedback. For this project, the sprint length was determined to be two weeks.

Each sprint consisted of the following:

- Two independent team meetings
- One class meeting
- One or more client meetings
- Stand-ups at the start of each meeting
- Sprint planning meetings
- Story pointing

The project backlog is tracked on Yodiz. Stories on Yodiz were broken into tasks which were assigned to a development and quality assurance teams of each sprint. Story pointing was conducted on planning poker.

Sprint Planning Meeting

The sprint planning meeting is a collaborative effort by the whole agile team where the product backlogs belonging to each sprint are assigned story points and members responsible to complete each task. The team used planning poker to assign point values to each task. Larger tasks were assigned to teams of individuals. The sprint planning session is headed by the scrum master and the entire agile team who determine the necessary effort needed for each task.

Scrum/Stand Up

The SCRUM stand-up was conducted at each of the team meetings. Scrum, or a stand up, is done within the team and is headed by the scrum master. Scrum is generally done in the first 15 minutes of the meeting before the team starts working on any of the tasks. Each of the team members is expected to answer the questions below:

- What did you do last week?
- What will you do this week?
- Are there any impediments?

Story Pointing (Estimation)

During each sprint planning session, all the members in the Agile team get a chance to assign story points for each of the tasks in the product backlog. Story pointing a task is assigning an estimate of the time taken for each individual task to be completed. We as a team decided for 1 story point be equal to 8 hours of our individual work, 2 story points equal 16 hours and so on. To provide an unbiased estimation, a tool called PointingPoker was used. This tool allows each individual team member to assign a story point without knowing what the other team members have voted for. After all the team members estimate the scrum master reviews the numbers and decides on what point each story should be assigned with.

Individual Reflections

Reflections are an important part of the SCRUM methodology as they are used to monitor the progress of the entire project by tracking the issues and impediments an individual faced during the sprint. Measures are taken based up on the reflections in a session called spring retrospective after each sprint.

Team members in general addresses the following questions:

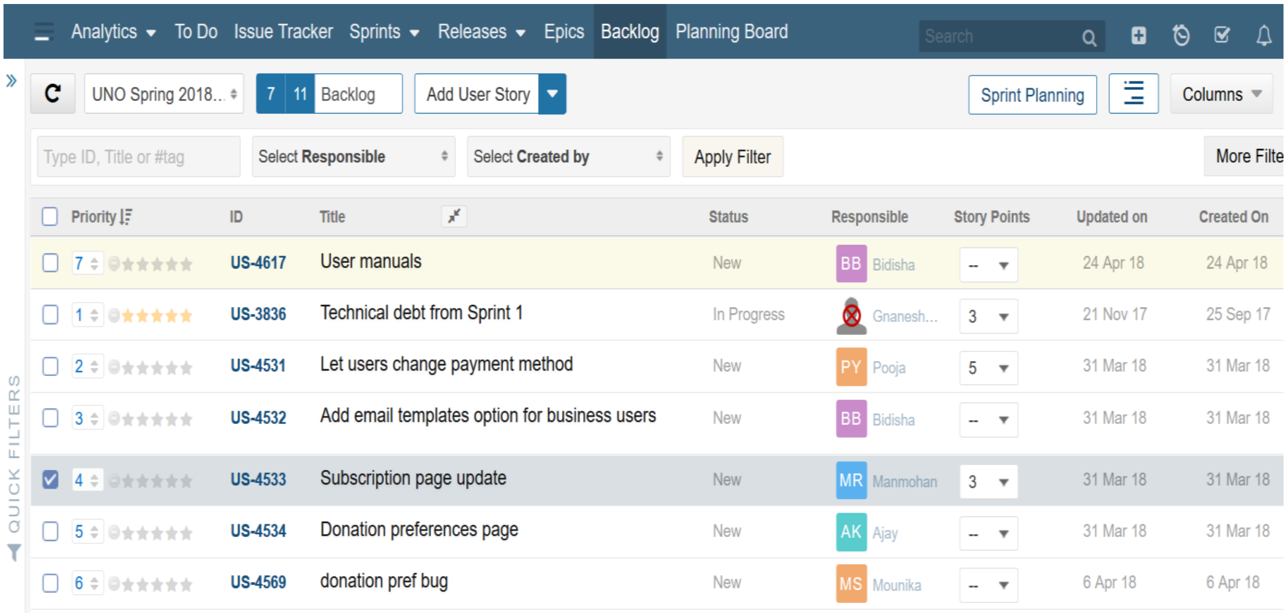
1. What went well in the sprint?
2. What went wrong in the spirit?
3. What lessons did I learn?

Project Management Tool

The team used a project management tool called Yodiz to track the progress of the project. Yodiz allows users capture project goals as epics, or features and then break these features into stories and stories into individual tasks. The time expected to be spent on each task defines the expected project velocity, or rate of completion by the team.

Product Backlog

The stories that compose each feature are added to the product backlog. The client liaison team must keep an eye on tasks and stories residing in the product backlog to ensure the client's priorities are reflected in the final product. An example of a product backlog is given below:



	Priority	ID	Title	Status	Responsible	Story Points	Updated on	Created On
□	7	US-4617	User manuals	New	BB Bidisha	--	24 Apr 18	24 Apr 18
□	1	US-3836	Technical debt from Sprint 1	In Progress	Ganesh...	3	21 Nov 17	25 Sep 17
□	2	US-4531	Let users change payment method	New	PY Pooja	5	31 Mar 18	31 Mar 18
□	3	US-4532	Add email templates option for business users	New	BB Bidisha	--	31 Mar 18	31 Mar 18
☑	4	US-4533	Subscription page update	New	MR Manmohan	3	31 Mar 18	31 Mar 18
□	5	US-4534	Donation preferences page	New	AK Ajay	--	31 Mar 18	31 Mar 18
□	6	US-4569	donation pref bug	New	MS Mounika	--	6 Apr 18	6 Apr 18

Figure 5: Product backlog

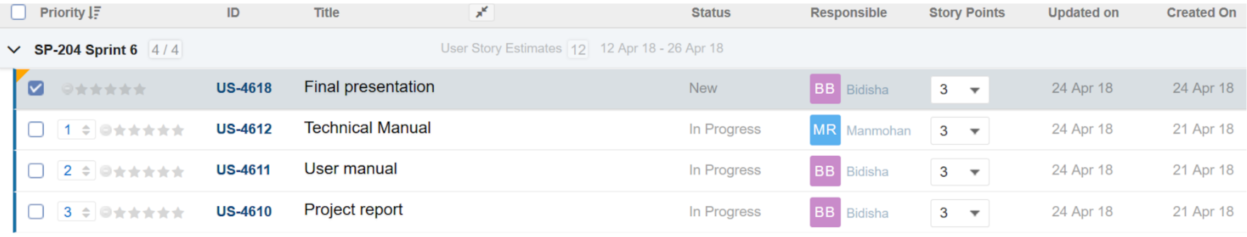
Schedule of Sprint release

Sprint	Duration	Tasks/Deliverables
0	Jan 18, 2018 - Feb 1 2018	Demo with client on 2/1/2018 Project preparedness: ● Statement of work ● Setting up local environment ● Verifying existing features ● User stories ● Mockups for Sprint 1
1	Feb 2, 2018 - Feb 15 2018	Demo with Client on 2/15/2018 Deliverables: ● Update currency format ● Allow business to update location from Business Profile ● Modify routes for Business Profile webpage ● Add a button “Invite code” on header to generate URL ● Encrypt URL key for additional security ● Request pending count corrected ● Fixed the ability display inactive business status in the dashboard ● Mockups for Sprint 2
2	Feb 16, 2018 - Mar 1 2018	Client Demo on 3/1/2018 Deliverables: ● Use 4 Status codes for CharityQ dashboard (Active, Cancel, Incomplete and Declined) ● Subscription plans will be visible and revamped with new UI ● Remove “Business admin” and “Business user” roles. ● Keep the color schemes as per client request (black and grey) on donation request page ● Change font to Semi-condensed Barlow for website ● Mockups for Sprint 3
3	Mar 2, 2018 - Mar 15, 2018	Client presentation on 3/15/2018 Deliverables: ● Review the message on business’s page for inactive forms ● Modify CharityQ admin email templates to substitute the first name only

		• Allow business to be in “Incomplete” status before they make payment • Modify the subscription such that businesses can use CharityQ services for remaining number of days once they cancel the subscription • Update Donation Preference Page to make more user friendly
4	Mar 16, 2018 - Mar 29, 2018	Client presentation on 3/29/2018 Deliverables: • Add a button to add email templates for ChatityQ admin • Able to delete a user at CharityQ User Level • Able to delete a user at Business Level • Change the UI for payment page • Change “Register your Business” to “Sign up now to start your free trial”
5	Mar 30, 2018 - Apr 12, 2018	Client presentation on 4/12/2018 Deliverables: • Allow Business admins to update their payment preferences • In the "Search donation" page, display columns as "Requested amount" and "Donated amount" • Testing for all the functionality • Bug fix on upcoming errors
6	Apr 13 - Apr 26, 2018	Client presentation on 4/26/2018 Deliverables: • Demo with client • Deliverables • Defect fixes • Finalize Documentations
	Between April 26 and May 3 2018	Final presentation with client on 5/03/2018 Deliverables: ● Working application ● Demonstration of all the features as promised in SOW ● Documentation: ○ User manual ○ Technical manual ○ Final project report

Sprint backlog

The sprint backlog consists of the list of tasks identified by the Scrum team to be delivered in the associated Scrum sprint. During the sprint planning meetings, all team members, including the scrum master and product owner work together to agree to what will be worked on during the next sprint (Martinelli and Milosevic, 2017). The team selects some number of product backlog items, in the form of user stories, and identify the tasks necessary to complete each user story. Each Sprint iteration for this project lasted 2 weeks.



□ Priority	ID	Title	Status	Responsible	Story Points	Updated on	Created On
SP-204 Sprint 6 4 / 4		User Story Estimates 12	12 Apr 18 - 26 Apr 18				
☑	US-4618	Final presentation	New	BB Bidisha	3	24 Apr 18	24 Apr 18
□ 1	US-4612	Technical Manual	In Progress	MR Manmohan	3	24 Apr 18	21 Apr 18
□ 2	US-4611	User manual	In Progress	BB Bidisha	3	24 Apr 18	21 Apr 18
□ 3	US-4610	Project report	In Progress	BB Bidisha	3	24 Apr 18	21 Apr 18

Figure 6: Sprint backlog list for Sprint 6

Definition of done

This term is used to describe that the completed user story has met the list of criteria below:

1. All Acceptance Criteria of the user story are met.
2. Code meets general Coding Standards.
3. Code is committed in version control system.
4. Functional tests are performed.
5. Code review is conducted
6. The code is covered by a minimum of 70% Unit Tests.
7. Integration tests of the affected areas are conducted and passed
8. All Unit Tests, Automated Acceptance tests have passed successfully.
9. Regression testing has been performed.
10. No Critical or Blocker bug exists in the bug backlog.
11. All documentation of technical and business nature has been approved and reviewed.

In simpler words, a story is done when we can show it to the intended parties to be a shippable feature.

Burn Down chart

"The Burndown chart is a graphical representation of the amount of work completed over a defined period of time as compared to the amount of work initially planned for completion" (Martinelli and Milosevic, 2017). The burn down chart was useful to provide a relatively accurate estimate of the team's progress. The burndown chart has time represented by x-axis as time and story points represented by y-axis. There is also an "ideal" burndown which is a straight line from top-left to bottom-right. The actual burndown is also to know the progress of the project. Each sprint has its own burn down chart. If the burn down chart shows that expected performance is behind schedule, the chart can be used as a real-time indicator, so the required corrective actions can be taken.

Yodiz gives two types of burndown charts –

- Burndown type - Point/Estimation Hours (User Stories): Blue line is projected burndown. It is calculated so that total points are divided by duration of sprint. Red line is calculated for story points that are completed. (Yodiz Blog, n.d.) Below is the example of the Burn Down Chart for Sprint 5:

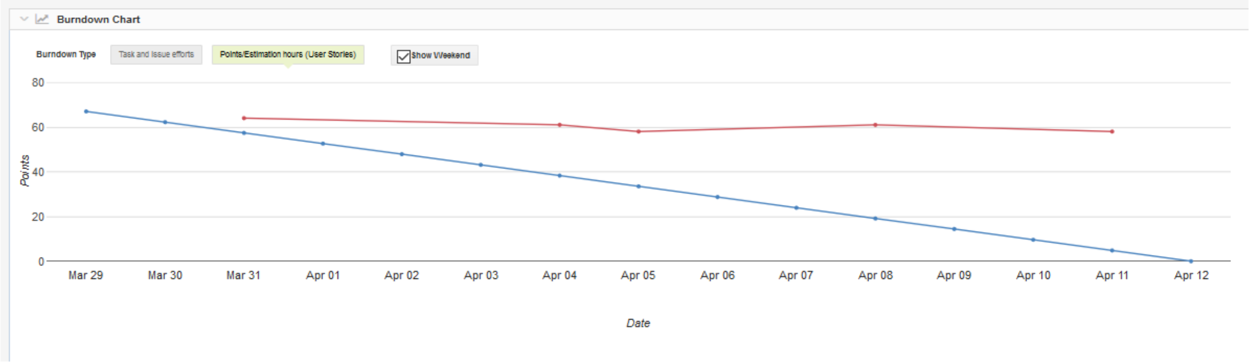


Figure 7: Burn down chart Point/Estimation Hours (User Stories):

- Burndown Type - Task and issue efforts: Blue line is projected burndown. It is calculated so that total hours are divided by duration of sprint. Red line is calculated for hour for Issues & Tasks that are spent. (Yodiz Blog, n.d.). Below is the example of the Burn Down Chart for Sprint 6:

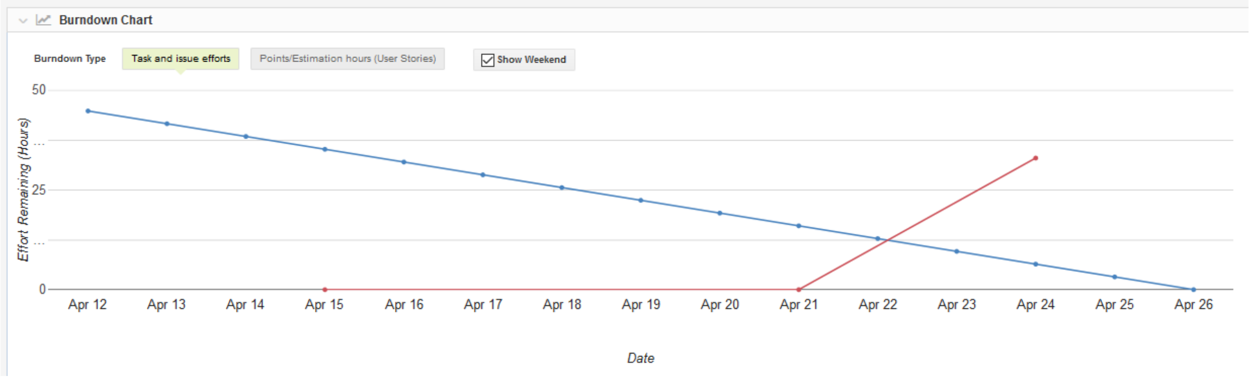


Figure 8: Burndown Type - Task and issue efforts

Project Velocity Dashboard

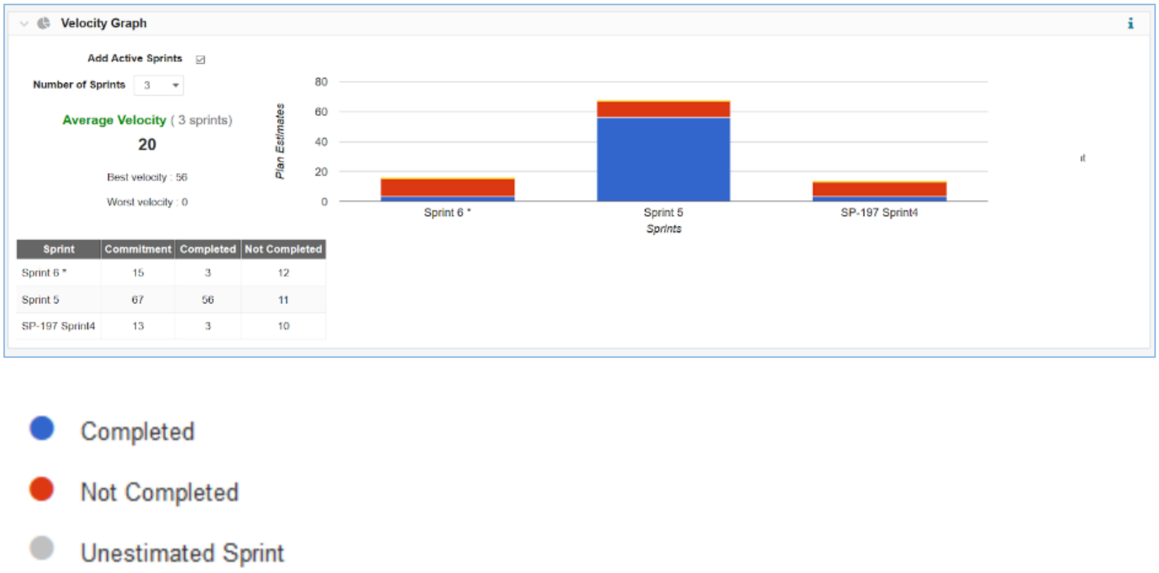
The Project Velocity Dashboard displays the Sprint Velocity (Velocity is based upon rate of completion of User Stories) which is useful for an in-depth analysis to deduce the rate at which User Stories are being completed. It can be used it to obtain the average velocity value for 3, 5 or 10 recently completed sprints. It helped the team efficiently utilize time effectively and plan for improvements.

Average Velocity: It is calculated by dividing (completed story points)/(number of sprints)

Best Velocity: Velocity of best performing sprint (from last selected number of sprints)

Worst Velocity: Velocity of worst performing sprint (from last selected number of sprints)

Figure 7 is an example of our team velocity for the last three sprints:



Velocity Graph

Add Active Sprints ☑

Number of Sprints 3

Average Velocity (3 sprints)

20

Best velocity : 56

Worst velocity : 0

Sprint	Commitment	Completed	Not Completed
Sprint 6 *	15	3	12
Sprint 5	67	56	11
SP-197 Sprint4	13	3	10

Figure 9: Team velocity for Sprints 4-6

Application Architecture

This Laravel project follows MVC (Model-View-Architecture) architectural pattern, comprises separation of business logic from presentation associated with GUI.

Three components of MVC pattern:

- **Model**: It is about the data related logic that a user works with. The data being transferred between view and controller. It enforces all the business rules on the related data, by putting the implementation of business rules in model, the developer can ensure that nothing in the application can create invalid data.
- **View**: It comprises UI logic of the application. For instance, UI components like textbox, checkboxes, etc., that a user interacts with. It never handles incoming data but displays it once available.
- **Controller**: It acts as interface between Model and View to process all the inputs. It acts on the model and decide what action to perform next. It can be anything such as rendering a view or redirecting to another page.

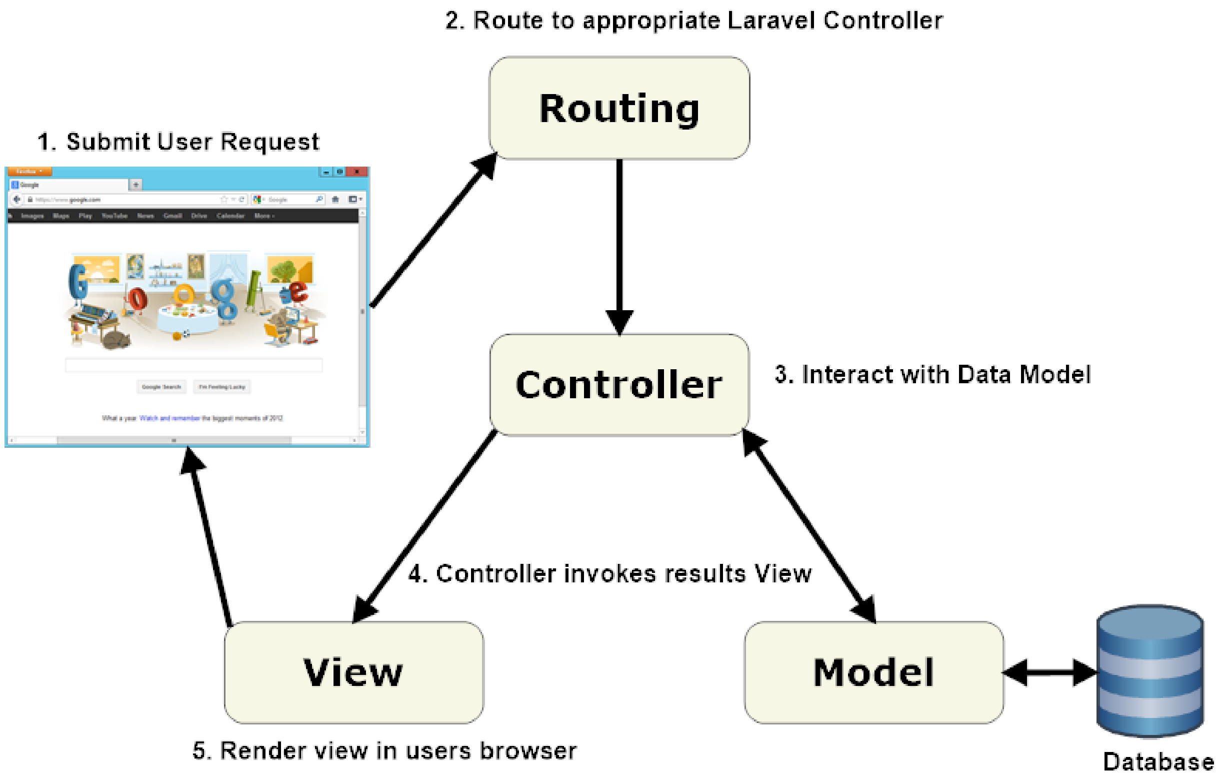

2. Route to appropriate Laravel Controller
Routing
1. Submit User Request
Controller
3. Interact with Data Model
4. Controller invokes results View
View
Model
Database
5. Render view in users browser

Figure 10: Model-View-Controller representation

Technical Overview, Tools, and Standards

Technical Team Overview

We are a small team of 8 graduate students. Hence irrespective of our individual leadership roles, we had all taken part in the technical objectives of the project to make it a success. For of the 8 team members have taken active part in the development, two of us has been dedicated to testing. The testers ensured that they work in tandem with the developers. Simultaneously the regression and functional testing has been done. The remaining two students were responsible for majority of the documentation. Before each sprint, stories were created and logged in Yodiz. Each story was accompanied by its associated tasks and a pointer indicating its complexity. The active developers were responsible for both the full stack development which consisted of both the back-end and front-end development. The passive developers had taken up the tasks of improving UI, or fix the bugs raised by the testers. In later stages, the team meetings had included combined effort of the team to bring out changes that are required to bring uniformity to the website.

Technical Tools

The technical team has used the below tools and frameworks**:**

Server

- PHP: Version 7.0 or newer
- Apache Server: Version 2.4.23
- MySQL: Version 5.7.14 or newer

Framework

- Laravel: Version 5.5 or newer

Version Control

- GitHub Desktop: Version 1.0.9 or newer
- GitHub: Version 2.11 or newer

Development Environment

- PHPStorm: Version 2017.3 or newer

Qualitative Analysis

- Selenium: Version 3.8.1

Technical Standards

Both the Development team and the QA team have followed the best practices used in large organizations in our project. Some of them are

- Having comments for the code being written
- Having proper indentation for the code being written
- Using the same story to report any bugs and resolving them
- Assigning a tester for each user story

Quality Management, Acceptance and Testing

Quality Management

Quality Assurance (QA) team made sure to test the features and report to the development team if any bugs or errors were identified and ensure that the end product satisfies all the functionalities mentioned in Statement of Work document signed by the client.

QA team was responsible for:

- Ensuring that the business requirements were incorporated in the developed web-application
- Writing test cases
- Manual testing (smoke testing, system testing, regression testing)
- Reporting and retesting bugs
- Ensuring bug fixes to be robust
- Automation Testing
- Creating and maintaining automation test scripts
- Helping Dev Team to fix small issues.

Acceptance Criteria & Testing

Acceptance Criteria are a set of rules that specify functional requirements applicable at the current phase of project development. These Criterias are set up in Yodiz during the initial sprint sessions of assigning user stories. There are no partial methods to define Acceptance Criteria, it is either considered “Done” or “Failed”.

Test Cases

Test cases were written by the Testing team in consultation with the Client liaison and were placed in a shared drive. The Test Lead assigned the test cases amongst the testers and was reviewed by all the project members. The format for writing the test case was designed and shared with the team. The individual test cases had all the necessary steps to verify that the code was working in the desired manner. It included space for documenting both the expected results as well as any bugs that the testing team encountered. The test cases were designed by taking many factors into consideration such as business analysis, user acceptance criteria and inputs from Product Owner.

Use of Yodiz to Certify Compliance/Functionality

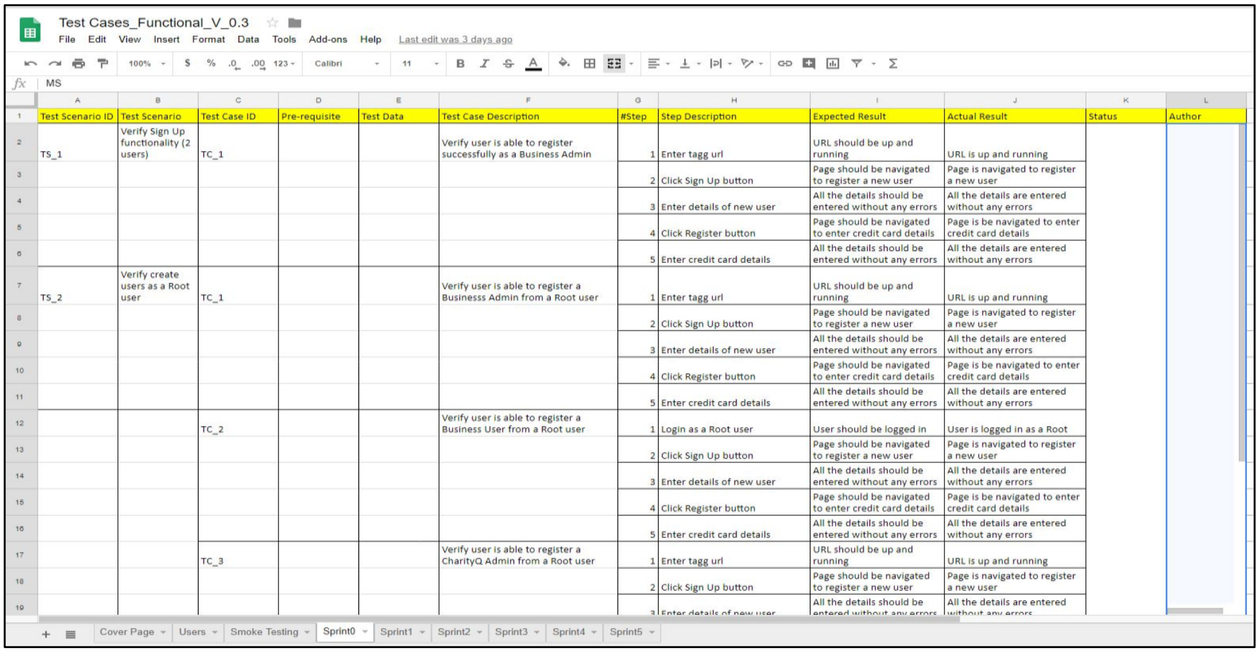
Yodiz was used as a channel for communication between the testing team and the development team. Yodiz was updated by scrum master during a planning session at the beginning of each sprint. The Quality Assurance team

Manual Test Cases

Manual test cases helped team in following aspects:

- Streamlined tracking of the feature implementation
- Structured communication between the QA and development teams
- Helped with the distribution of work within the QA team.

Following is the template used (Figure 11):



	A	B	C	D	E	F	G	H	I	J	K	L
1	Test Scenario ID	Test Scenario	Test Case ID	Pre-requisite	Test Data	Test Case Description	#Step	Step Description	Expected Result	Actual Result	Status	Author
2	TS_1	Verify Sign Up functionality (2 users)	TC_1			Verify user is able to register successfully as a Business Admin	1	Enter tagg url	URL should be up and running	URL is up and running		
3							2	Click Sign Up button	Page should be navigated to register a new user	Page is navigated to register a new user		
4							3	Enter details of new user	All the details should be entered without any errors	All the details are entered without any errors		
5							4	Click Register button	Page should be navigated to enter credit card details	Page is be navigated to enter credit card details		
6							5	Enter credit card details	All the details should be entered without any errors	All the details are entered without any errors		
7	TS_2	Verify create users as a Root user	TC_1			Verify user is able to register a Businesss Admin from a Root user	1	Enter tagg url	URL should be up and running	URL is up and running		
8							2	Click Sign Up button	Page should be navigated to register a new user	Page is navigated to register a new user		
9							3	Enter details of new user	All the details should be entered without any errors	All the details are entered without any errors		
10							4	Click Register button	Page should be navigated to enter credit card details	Page is be navigated to enter credit card details		
11							5	Enter credit card details	All the details should be entered without any errors	All the details are entered without any errors		
12			TC_2			Verify user is able to register a Business User from a Root user	1	Login as a Root user	User should be logged in	User is logged in as a Root		
13							2	Click Sign Up button	Page should be navigated to register a new user	Page is navigated to register a new user		
14							3	Enter details of new user	All the details should be entered without any errors	All the details are entered without any errors		
15							4	Click Register button	Page should be navigated to enter credit card details	Page is be navigated to enter credit card details		
16							5	Enter credit card details	All the details should be entered without any errors	All the details are entered without any errors		
17			TC_3			Verify user is able to register a CharityQ Admin from a Root user	1	Enter tagg url	URL should be up and running	URL is up and running		
18							2	Click Sign Up button	Page should be navigated to register a new user	Page is navigated to register a new user		
19							3	Enter details of new user	All the details should be entered without any errors	All the details are entered without any errors		

Figure 11: Test cases template

Automation Test Scripts

Application is being tested using Selenium automation. Regression scenarios uses automation and reduces the efforts of testing. Browser session is invoked initially using these test scripts and automation enables testing various scenarios. TestNG feature is used in the testing framework to create assertions and validate the application behavior. The results generated by the test scripts are used to pass manual test cases for functional testing and Regression test suite.

Selenium Automation Framework

As a QA team for the project, we took responsibility to develop an automation framework to run the entire application in an automated way rather than manual. This gives confidence in the client to verify the working functionality of the application and make sure application is running according to the business requirements.

The following testing scenarios are used.

Smoke Testing

Smoke Testing was carried out by the QA team immediately after the test environment is available to test the application. This ensures system stability and the existing application loopholes.

Integration testing

Integration testing was performed after the combining individual feature developments. At this stage, QA with the direction of Dev will perform this testing. This ensures that all modules of the application are working as per business integration flow.

Regression testing

Regression testing is performed on the application rite from completion of Sprint 1 testing. After every sprint development, a basic regression suite is run to test the existing functionality. A functional regression suite is created for current sprint with new functionalities to run as basic regression in next sprint.

User Acceptance Test (UAT)

User Acceptance Test (UAT) was performed by the Client and Product Owner in Client location. Entire application was tested in the perspective of the end user. UAT ensured the client that all the features promised in SOW document are delivered as expected. After Client reviewing and confirming that all the test requirements are implemented in the project, a sign-off will be provided by the team to the project.

Project Retrospective

What went right?

It was a great learning experience. We got a chance to apply the skills we have acquired in the since the beginning of our graduate studies. It as an amazing team work. Every person has a strengths and weakness. A good team is the one that combines the strengths of everyone and shadows the weakness. It made us understand how the strength of every individual within a team contributes towards the success of a project. Towards the end of the project, we can rightly conclude that we were a good team that stood strong all-throughout the project.

What went wrong?

At times there were difference in opinions. On certain sprints we were unable to provide the deliverables entirely. But ultimately, we not only accomplished the promised deliverables but also the nice-to-haves (features that are mentioned as out of scope in the Statement of Work).

Lessons Learned

We have gained some leadership qualities and hands-on experience on technologies. We realized the importance of best practices in terms of coding and documentation and the core principles of Agile development methodologies.

Project Challenges

- Development Learning Curve
    - Best practices
    - Code quality
    - Merging issues

- Team Building
    - Time Management
    - Accountability in Yodiz

Conclusion

This capstone class started as a group of strangers, and over the course of a few week came together to become a team. We honed skills we did not realize we had. We understood the essence of collaboration and teamwork. We can proudly deliver a production ready website to our clients. We helped each other with technology, time management, and communication skills. This project has allowed us to develop a better understanding of both the technology used in projects, and team building resources.