

PPA Membership Maintenance System

Sprint Report 1

Capstone Computing Project 2

Group SD07

Semester 2, 2018

Curtin University – Department of Computing

Assignment Cover Sheet / Declaration of Originality

Complete this form if/as directed by your unit coordinator, lecturer or the assignment specification.

Last name:	Hingalagoda	Student ID:	19211804
Other name(s):	Bhanuka		
Unit name:	Capstone Computing Project 2	Unit ID:	ISAD3001
Lecturer / unit coordinator:	Dr. Hannes Herrmann	Tutor:	Ms. Geethanjalie Wimalarathne
Date of submission:	17/08/2018	Which assignment?	Sprint Report 01

I declare that:

- The above information is complete and accurate.
- The work I am submitting is *entirely my own*, except where clearly indicated otherwise and correctly referenced.
- I have taken (and will continue to take) all reasonable steps to ensure my work is *not accessible* to any other students who may gain unfair advantage from it.
- I have *not previously submitted* this work for any other unit, whether at Curtin University or elsewhere, or for prior attempts at this unit, except where clearly indicated otherwise.

I understand that:

- Plagiarism and collusion are dishonest, and unfair to all other students.
- Detection of plagiarism and collusion may be done manually or by using tools (such as Turnitin).
- If I plagiarise or collude, I risk failing the unit with a grade of ANN ("Result Annulled due to Academic Misconduct"), which will remain permanently on my academic record. I also risk termination from my course and other penalties.
- Even with correct referencing, my submission will only be marked according to what I have done myself, specifically for this assessment. I cannot re-use the work of others, or my own previously submitted work, in order to fulfil the assessment requirements.
- It is my responsibility to ensure that my submission is complete, correct and not corrupted.

Signature:  _____ Date of signature: 17/08/2018

(By submitting this form, you indicate that you agree with all the above text.)

Table of Contents

1. Introduction	2
1.1 Group Introduction	2
1.2 Project Introduction.....	2
2. Progress Update.....	3
2.1 Allocated Tasks for the Sprint 1	3
2.2 Planned Tasks for the Sprint 2	3
2.3 Difficulties	4
3. Task Break Down.....	5
3.1 Task PPA-16 Prepare System Feature 4	5
3.2 Task PPA-18 Prepare Software and Communications Interfaces	5
3.3 Task PPA-18 Prepare Other Nonfunctional Requirements	5
3.4 Task PPA-25 Requirement Gathering.....	6
3.5 Task PPA-27 Requirements Analyzing.....	6
3.6 Task PPA-32 Learn MEAN Stack	7
1. Sub Task Learn Mongo DB	7
2. Sub Task Learn Node JS and Express JS.....	7
3.7 Task PPA-37 Sprint 2 Planning	11
3.8 Task PPA-42 Find a suitable Database System.....	12
3.9 Task PPA-62 Research about similar systems	12
3.10 Task PPA-48 Setting up the individual development environment.....	12
3.11 Task PPA-56 Task Allocation Planning	13
4. Development Methodology.....	14
4.1 Minutes	14
4.2 Burndown Chart.....	15
4.3 Sprint Retrospective.....	15
4.4 Task Summary	16
4.5 Time Management.....	16

1. Introduction

1.1 Group Introduction

All the members in our group have successfully completed the CCP1 in 2017. We enrolled for the CCP2 module in 2018 2nd semester, therefore this project is started in semester 2 of this year (2018). Because of that we have to done the main documentations and tasks of the project such as SRS, task allocation, initial requirement gathering etc. in semester 2. Each of the group members has to do a workload of 2 semesters within this semester, in order to complete the project successfully.

1.2 Project Introduction

PPA Membership Maintenance System is going to be used for membership management and some other important administration tasks such as event planning, donation collecting, accounts handling etc. of past pupil association of Sirimavo Bandaranaike Vidyalaya. Currently all these operations are manually performed by the committee. The main intention of the system is to automate most of those tasks and perform the semi-automated tasks easily and conveniently.

We use MEAN stack to develop this application, JIRA as the project management tool and bitbucket as the online repository. The application has main 4 parts as Membership Services, Accountings, Event Planning and Reporting. These four sections are interconnected with each other as per their functionalities.

2. Progress Update

Sprint 1: 03rd August – 17th August

2.1 Allocated Tasks for the Sprint 1

#	Task ID	Task	Task Status	Hours Estimated
1	PPA-16	Prepare System Feature 4	Completed	1
2	PPA-18	Prepare Software and Communications Interfaces	Completed	1
3	PPA-19	Prepare Other Nonfunctional Requirements	Completed	2
4	PPA-25	Requirement Gathering	Completed	1
5	PPA-27	Requirements Analyzing	Completed	3
6	PPA-32	Learn MEAN Stack	Completed	6
7	PPA-37	Sprint 2 Planning	Completed	2
8	PPA-42	Find a suitable Database System	Completed	2
9	PPA-48	Setting up the individual development environment	Completed	1
10	PPA-56	Task Allocation Planning	Completed	5
11	PPA-62	Research about similar systems	Completed	2
		Total Hours		26

2.2 Planned Tasks for the Sprint 2

#	Task ID	Task	Estimation (Hours)
1	PPA-	Gallery Page Main GUI Designing	2
2	PPA-	Gallery Page Database Deigning	1
3	PPA-	Gallery Page Presentation Tier Logic	1
4	PPA-	Gallery Page Middle Tier Logic	2
5	PPA-	Gallery Page Testing	1
6	PPA-	Create Project Page - Draft GUI	2
7	PPA-	Create Project Page - Database Design	1
8	PPA-	Create Project Page - Middle Tier Logic	2
9	PPA-	Create Project Page - Presentation Tier Logic	1
10	PPA-	View Project Page - Draft GUI	2
11	PPA-	Project Timeline - Draft GUI	2
12	PPA-	Project Timeline - Middle Tier Logic	2
13	PPA-	Project Timeline - Presentation Tier Logic	2

14	PPA-	Project Timeline - Handling Photos and Videos	3
Total Hours			24

2.3 Difficulties

- There were some issues occurred when installing the angular framework. Please check the task breakdown for Task 4 for more information.

3. Task Break Down

3.1 Task PPA-16 Prepare System Feature 4

Estimate Time: 1 Hours

Actual Time: 1 Hours

Actual Time (this sprint): 1 Hours

Description

This is a sub task of preparing SRS Document task.

3.2 Task PPA-18 Prepare Software and Communications Interfaces

Estimate Time: 1 Hours

Actual Time: 1 Hours

Actual Time (this sprint): 1 Hours

Description

This is a sub task of preparing SRS Document task. In this task I Completed the Software and Communications Interfaces section of the SRS document.

3.3 Task PPA-18 Prepare Other Nonfunctional Requirements

Estimate Time: 1 Hours

Actual Time: 1 Hours

Actual Time (this sprint): 1 Hours

Description

This is a sub task of preparing SRS Document task. In this task I Completed the Other Nonfunctional Requirements section that includes,

- Performance Requirements
- Safety Requirements
- Security Requirements
- Software Quality Attributes

of the SRS document.

3.4 Task PPA-25 Requirement Gathering

Estimate Time: 1 Hours

Actual Time: 1 Hours

Actual Time (this sprint): 1 Hours

Description

This is a task is to meet the client and gather requirements for the project.

The meeting minutes can be found in the link below.

[Client meeting 1 minutes](#)

3.5 Task PPA-27 Requirements Analyzing

Estimate Time: 3 Hours

Actual Time: 3 Hours

Actual Time (this sprint): 3 Hours

Description

This is a task is to meet the client and gather requirements for the project.

The meeting minutes can be found in the link below.

[Client meeting 1 minutes](#)

3.6 Task PPA-32 Learn MEAN Stack

1. Sub Task Learn Mongo DB

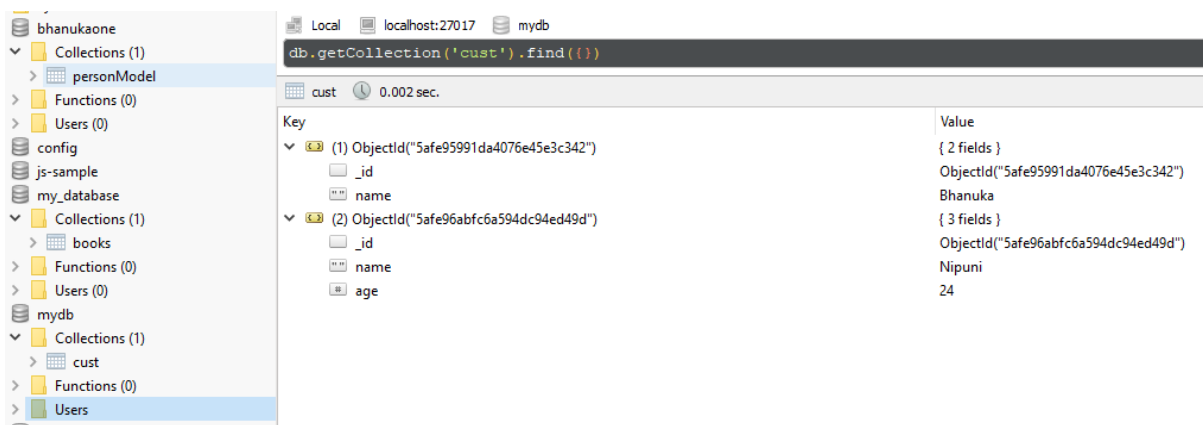
Estimate Time: 2 Hours

Actual Time: 2 Hours

Actual Time (this sprint): 2 Hours

Description

This is a task is to meet the client and gather requirements for the project.



2. Sub Task Learn Node JS and Express JS

Estimate Time: 6 Hours

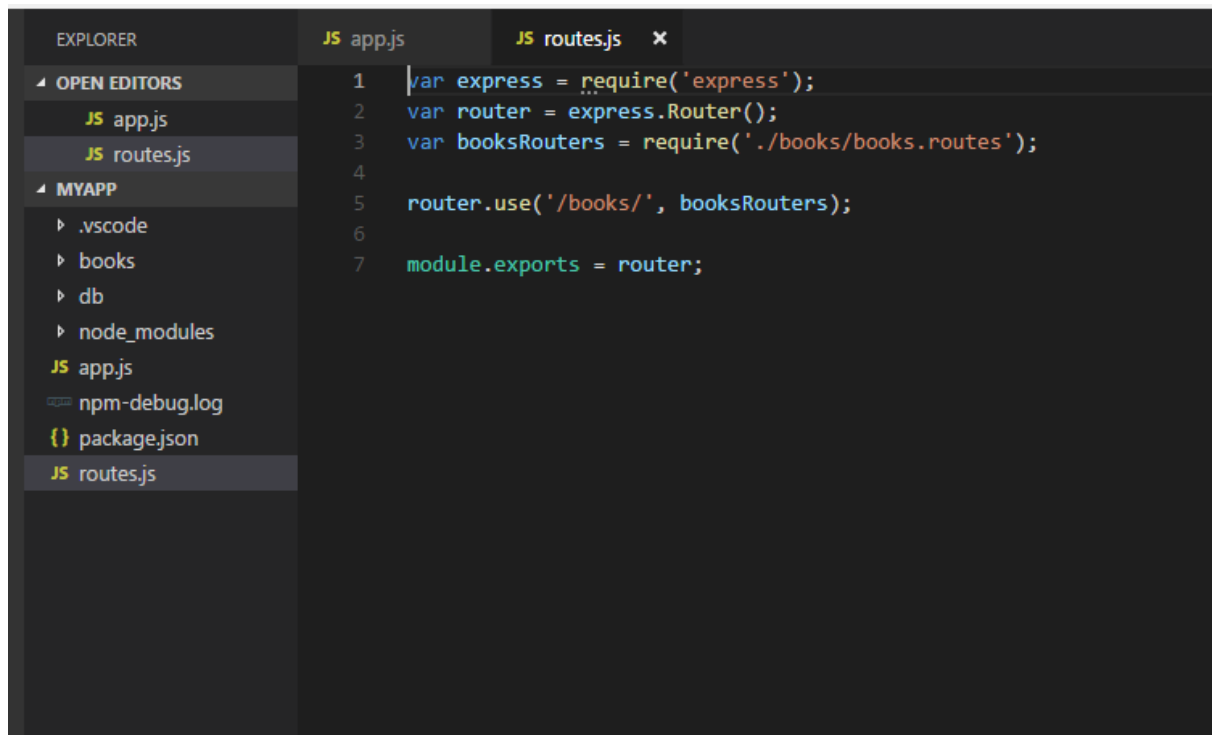
Actual Time: 6 Hours

Actual Time (this sprint): 6 Hours

Description

Learning from udeimi course.

Proof



EXPLORER

JS app.js JS routes.js

OPEN EDITORS

JS app.js JS routes.js

MYAPP

- .vscode
- books
- db
- node_modules

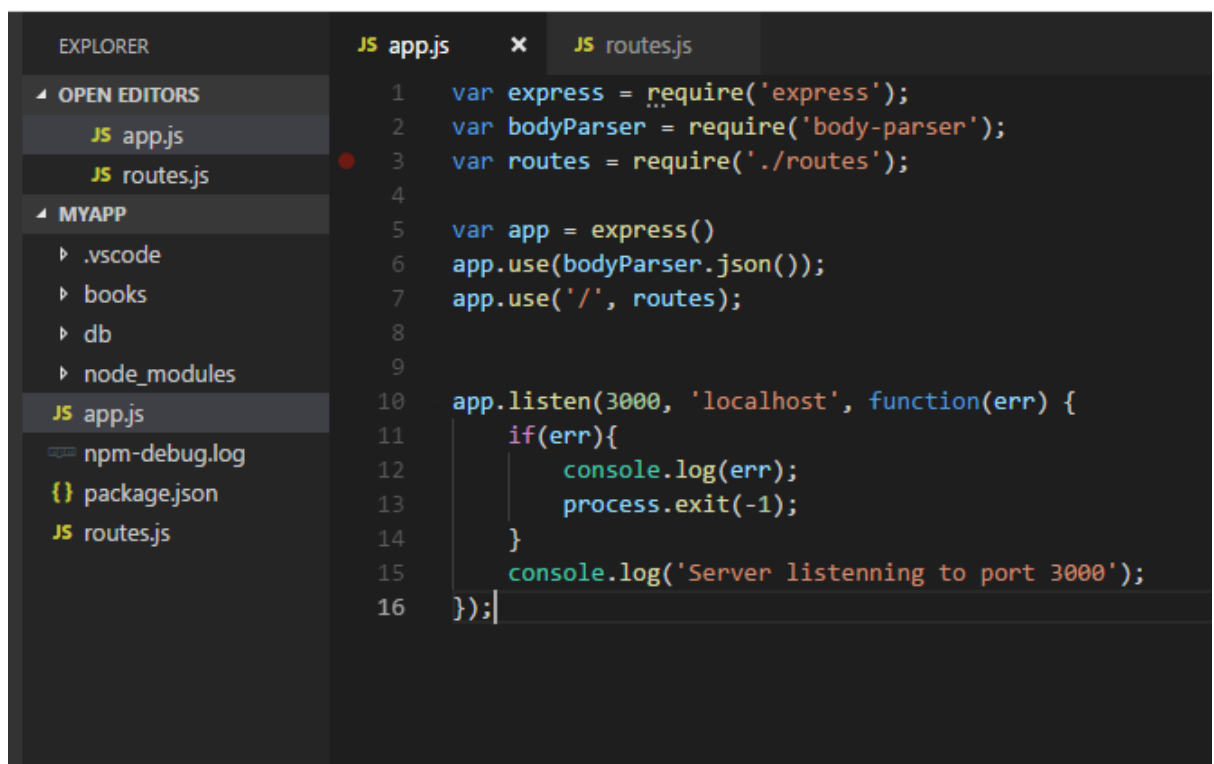
JS app.js

npm-debug.log

package.json

JS routes.js

```
1 |var express = require('express');
2 |var router = express.Router();
3 |var booksRoutes = require('./books/books.routes');
4
5 |router.use('/books/', booksRoutes);
6
7 |module.exports = router;
```



EXPLORER

JS app.js JS routes.js

OPEN EDITORS

JS app.js JS routes.js

MYAPP

- .vscode
- books
- db
- node_modules

JS app.js

npm-debug.log

package.json

JS routes.js

```
1 |var express = require('express');
2 |var bodyParser = require('body-parser');
3 |var routes = require('./routes');
4
5 |var app = express()
6 |app.use(bodyParser.json());
7 |app.use('/', routes);
8
9
10 |app.listen(3000, 'localhost', function(err) {
11 |    if(err){
12 |        console.log(err);
13 |        process.exit(-1);
14 |    }
15 |    console.log('Server listenning to port 3000');
16 |});
```

JS app.js

JS db.config.js x

JS routes.js

```
1  const mongoose = require('mongoose');
2
3
4  const schema = mongoose.Schema;
5
6  const book = new schema({
7    name: {
8      type: String,
9      require: true
10   },
11   author: {
12     type: String,
13     require: true
14   }
15 });
16
17 mongoose.model('Books', book);
18
19 mongoose.connect('mongodb://localhost:27017/my_database', function(err){
20   if(err){
21     console.log(err);
22     process.exit(-1);
23   }
24   console.log('Connected to db');
25 });
26
27
28
29 module.exports = mongoose;
```

JS app.js JS db.config.js x JS books.controller.js x JS routes.js

```
1 var mongoose = require('../db/db.config');
2 const bookSchema = mongoose.model('Books');
3
4 var bookController = function(){
5   this.searchAll = function(){
6     return new Promise(function(resolve, reject){
7       bookSchema.find().exec().then(function (data){
8         resolve({
9           status: 200,
10          data: data
11        })
12      }).catch(function(err){
13        reject({
14          status: 404,
15          message: "No data"
16        })
17      })
18    })
19  }
20
21  this.addBook = function(data){
22    return new Promise(function(resolve, reject){
23      newBook = new bookSchema({
24        name: data.name,
25        author: data.author
26      });
27      newBook.save().then(function(){
28        resolve({
29          status: 200,
30          message: "Success!"
31        });
32      }).catch(function(err){
33        reject({
34          status: 500,
35          message: "Error - " + err
36        })
37      })
38    })
39  }
```

```
JS app.js JS db.config.js JS books.controller.js JS books.routes.js JS routes.js
1 var express = require('express');
2 var bookController = require('./books.controller');
3 var router = express.Router();
4
5 router.get('/', function (req, res) {
6     bookController.searchAll().then(function(data){
7         res.status(data.status).send({
8             data: data.data
9         });
10    }).catch(function(err){
11        res.status(err.status).send({
12            message: err.message
13        })
14    })
15 });
16
17 router.post('/', function (req, res) {
18     bookController.addBook(req.body).then(function(data){
19         res.status(data.status).send({
20             message: data.message
21         });
22     }).catch(function(err){
23         res.status(data.status).send({
24             message: data.message
25         })
26     })
27 });
28
29 router.get('/:id', function (req, res) {
30     bookController.search(req.params.id).then(function(data){
31         res.status(data.status).send({
32             data: data.data
33         });
34     }).catch(function(err){
35         res.status(err.status).send({
36             message: err.message
37         })
38     })
39 });
```

3.7 Task PPA-37 Sprint 2 Planning

Estimate Time: 2 Hours

Actual Time: 1 Hours

Actual Time (this sprint): 1 Hours

Description

This is a task is to meet the client and gather requirements for the project.

3.8 Task PPA-42 Find a suitable Database System

Estimate Time: 2 Hours

Actual Time: 2 Hours

Actual Time (this sprint): 2 Hours

Description

Goal of this research to consider the functional and nonfunctional requirements of out project and find the best database management software that can easily cater those requirements.

More detailed document link -> [Database Research](#)

3.9 Task PPA-62 Research about similar systems

Estimate Time: 2 Hours

Actual Time: 2 Hours

Actual Time (this sprint): 2 Hours

Description

Goal of this research is to find existing Association Management Systems and list their features that can be implement in PPA Membership Maintenance System.

More detailed document link -> [Similar Systems Research](#)

3.10 Task PPA-48 Setting up the individual development environment

Estimate Time: 1 Hours

Actual Time: 1 Hours

Actual Time (this sprint): 1 Hours

Description

This task is to setup a development environment by installing required software.

3.11 Task PPA-56 Task Allocation Planning

Estimate Time: 5 Hours

Actual Time: 4 Hours

Actual Time (this sprint): 4 Hours

Description

This task is to divide workload to tasks and prepare task allocation document.

4. Development Methodology

4.1 Minutes

We use jira as a project management tool. Here is my issue dashboard.

The screenshot shows a Jira issue dashboard for the project 'PPA Membership Maintenance System'. The issue title is 'Prepare Other Nonfunctional Requirements' (PPA-19). The issue is a 'Sub-task' with a 'Medium' priority, 'None' labels, and is assigned to 'Bhanuka Hingalagoda'. The status is 'CLOSED' with a resolution of 'Done'. The issue is part of 'PPA Sprint 1'. The dashboard includes a sidebar with a list of issues, a main panel with details, description, and activity, and a right panel with people, dates, time tracking, and agile information.

Details:

- Type: Sub-task
- Priority: Medium
- Labels: None
- Sprint: PPA Sprint 1
- Status: CLOSED (View Workflow)
- Resolution: Done

Description:

Prepare Other Nonfunctional Requirements

- Performance Requirements
- Safety Requirements
- Security Requirements
- Software Quality Attributes

Activity:

All Comments Work Log History Activity

Bhanuka Hingalagoda logged work - 10/Aug/18 11:13 PM

Time Spent: 2m

<No comment>

Bhanuka Hingalagoda logged work - 10/Aug/18 11:15 PM

Time Spent: 1h 58m

<No comment>

People:

- Assignee: Bhanuka Hingalagoda
- Reporter: Bhanuka Hingalagoda
- Votes: 0
- Watchers: 1 Stop watching this issue

Dates:

- Created: 1 week ago 8:05 PM
- Updated: 1 week ago 11:28 PM
- Resolved: 1 week ago 11:16 PM

Time Tracking:

- Estimated: 2h
- Remaining: 0h
- Logged: 2h

Development:

- Create branch

Agile:

- Active Sprint: PPA Sprint 1 ends 17/Aug/18
- View on Board

Bitbucket Link for Client Meetings:

Meeting 1:

https://bitbucket.org/Computing_Projects_SLIIT/2018_sd07/src/master/Documents/Client%20Meeting%20Minutes/Client%20Meeting%201/SD07_Client_Meeting_1.pdf

Meeting 2:

https://bitbucket.org/Computing_Projects_SLIIT/2018_sd07/src/master/Documents/Client%20Meeting%20Minutes/Client%20Meeting%202/SD07_Client_Meeting_2.pdf

We had 10 standup meetings within this sprint and the minutes of those meetings can be found from the following link.

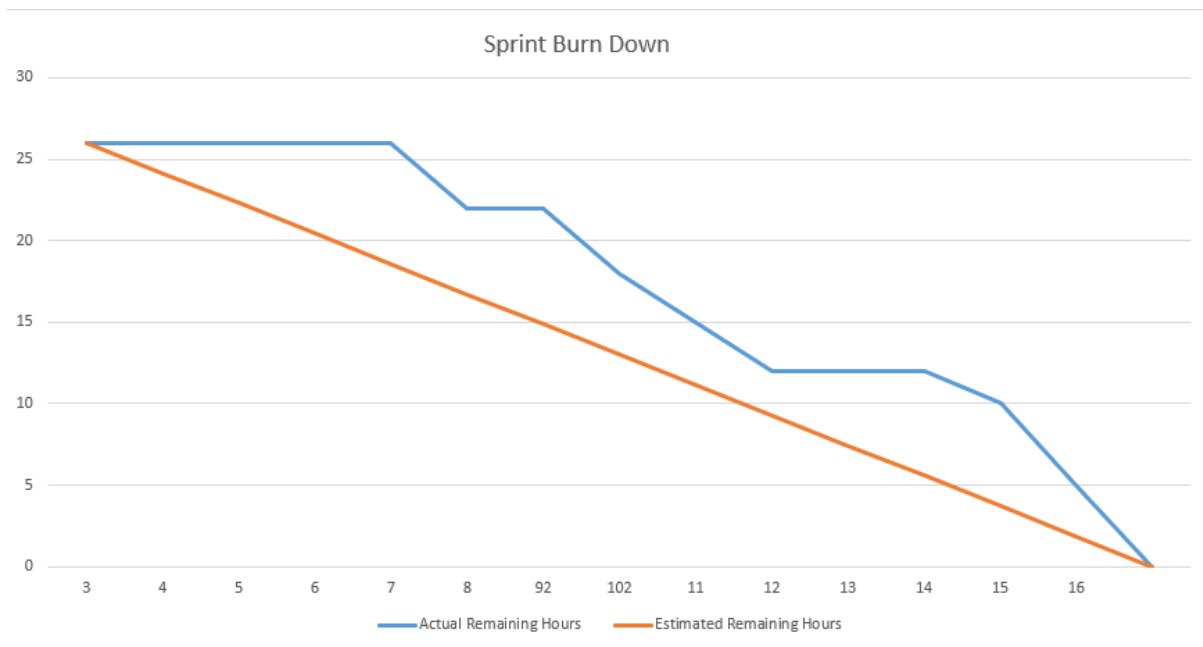
Standup Meeting Minutes:

https://bitbucket.org/Computing_Projects_SLIIT/2018_sd07/src/master/Documents/Standup%20Meeting%20Minutes/Sprint%201/Standup%20Meeting%20-%20Sprint%201.docx

4.2 Burndown Chart

Estimate Time: 26 Hours

Actual Time: 24 Hours



4.3 Sprint Retrospective

This sprint was tightly packed with many tasks since we have to properly understand project scope, prepare SRS, Task Allocation Document etc. which considered as semester 1 tasks. We had to do all that within this sprint because we started our project on semester 2. So we had 2 client meetings to clearly understand the client's requirements. Most of our standup meetings was held to discuss the proposed system functionalities and the way we should implement them. During the first week we roughly finalize the basic functionalities and prepared the SRS. During the second week of this sprint we had another client meeting (client meeting 2) and properly finalized all the system functionalities to prepare the task allocation document. We had to have longer discussion meetings to breakdown the functionalities into tasks and sub tasks. After the hours of discussions finally we prepared a draft document which represent the main functionalities with basic task breakdown and assigned the sections for each member of the team to further breakdown.

Basic Task Breakdown:

https://bitbucket.org/Computing_Projects_SLIIT/2018_sd07/src/master/Documents/Sprint%20Documents/Sprint%201/Tasks%20Breakdown.docx

Then we created a google sheet, so all of the members can work concurrently in there to prepare the TA. That went well, and we finished the TA document on the last day of this sprint.

On the last day of the sprint we planned the tasks for the sprint 2 and created the sprint 2 sprint board by adding them into the board. The most of the tasks in this sprint were planning based and documentation based tasks. But the most of the sprint 2 tasks are development based tasks.

4.4 Task Summary

Estimate Time: 26 Hours

Actual Time: 24 Hours

Completed 11 Task during this Sprint.

4.5 Time Management

#	Task ID	Task	Hours Estimated	Actual
1	PPA-16	Prepare System Feature 4	1	1
2	PPA-18	Prepare Software and Communications Interfaces	1	1
3	PPA-19	Prepare Other Nonfunctional Requirements	2	2
4	PPA-25	Requirement Gathering	1	1
5	PPA-27	Requirements Analyzing	3	3
6	PPA-32	Learn MEAN Stack	6	6
7	PPA-37	Sprint 2 Planning	2	1
8	PPA-42	Find a suitable Database System	2	2
9	PPA-48	Setting up the individual development environment	1	1
10	PPA-56	Task Allocation Planning	5	4
11	PPA-62	Research about similar systems	2	2
	Total		26	24