

PPA Membership Maintenance System

Sprint Report 4
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 Wimalaratne
Date of submission:	28/09/2018	Which assignment?	Sprint Report 04

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: 28/09/2018

(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 4	3
2.2 Planned Tasks for the Sprint 5	3
2.3 Difficulties	4
3. Task Break Down.....	5
3.1 Detailed Account Page Transaction Table Search Methods	5
3.2 Detailed Account Page Search & Filtering Method Calls	9
3.3 Task 3 Account Report Page GUI Designing.....	13
3.9 Task 9 Sprint 5 Planning	15
3.10 Task 10 Discuss design issues Sprint 4	16
4. Development Methodology.....	17
4.1 Minutes	17
4.2 Burndown Chart.....	17
4.3 Sprint Retrospective.....	18
4.4 Task Summary	18
4.5 Time Management.....	19

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 do 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 4: 15th September – 28th September

2.1 Allocated Tasks for the Sprint 4

#	Task ID	Task	Task Status	Hours Estimated
1	PPA-	Detailed Account Page Transaction Table Search Methods	Completed	3
2	PPA-	Detailed Account Page Search & Filtering Method Calls	Completed	2
3	PPA-	Account Report Page GUI Designing	Completed	2
4	PPA-	Account Report Page GUI Implementation	Completed	3
5	PPA-	Account Report Page GUI Report Generation Algorithm	Completed	4
6	PPA-	Research about report generation	Completed	4
7	PPA-	Service Letter Generation GUI Designing & Implementation	Completed	3
8	PPA-	Service letter generation Report Designing (PDF)	Completed	2
9	PPA-	Sprint 5 Planning	Completed	1
10	PPA-	Discuss design issues Sprint 4	Completed	1
Total Hours				25

2.2 Planned Tasks for the Sprint 5

#	Task ID	Task	Estimation (Hours)
1	PPA-	Letter Requests Insert Methods	1
2	PPA-	Letter Requests Get Methods	1
3	PPA-	Letter Requests Update Methods	1
4	PPA-	Letter Requests Delete Methods	1
5	PPA-	Member Directory GUI Design	1
6	PPA-	Member Directory GUI Implementation	3
7	PPA-	Member Directory GUI Validations	2
8	PPA-	Member Detail View GUI Implementation	4
9	PPA-	Member Detail Update Implementation	2
10	PPA-	Member Details Delete Implementation	2
11	PPA-	Member Details Search Functionality with Filtering Implementation	4
12	PPA-	Sprint 6 Planning	1
13	PPA-	Discuss design issues Sprint 5	1
Total Hours			24

2.3 Difficulties

3. Task Break Down

3.1 Detailed Account Page Transaction Table Search Methods

Estimate Time: 3 Hours

Actual Time: 3 Hours

Actual Time (this sprint): 3 Hours

Description

This task is to enable filtering to transaction table.

Front end code

```
<mat-form-field>
  <mat-select [(ngModel)]="filterOption" placeholder="Filter by">
    <mat-option value="all">All</mat-option>
    <mat-option value="account">Account ID</mat-option>
    <mat-option value="amount">Amount</mat-option>
    <mat-option value="desc">Description</mat-option>
    <mat-option value="entered">Entered By</mat-option>
    <mat-option value="donation">Donation</mat-option>
  </mat-select>
</mat-form-field>
<mat-form-field>
  <input matInput [(ngModel)]="filterString" placeholder="Filter">
</mat-form-field>
<mat-slide-toggle class="add-gap" [checked]="dtp1.enabled" (change)="dtp1.enabled ? disableDate() : enableDate()">
  Date Range
</mat-slide-toggle>
<mat-form-field class="add-gap">
  <input matInput [matDatepicker]="picker1" placeholder="Choose from date" [formControl]="dtp1">
  <mat-error *ngIf="dtp1.invalid">Please enter valid date.</mat-error>
  <mat-datepicker-toggle matSuffix [for]="picker1"></mat-datepicker-toggle>
  <mat-datepicker #picker1></mat-datepicker>
</mat-form-field>
<mat-form-field>
  <input matInput [matDatepicker]="picker2" placeholder="Choose until date" [formControl]="dtp2">
  <mat-error *ngIf="dtp2.invalid">Please enter valid date.</mat-error>
  <mat-datepicker-toggle matSuffix [for]="picker2"></mat-datepicker-toggle>
  <mat-datepicker #picker2></mat-datepicker>
</mat-form-field>
<button mat-icon-button (click)="onSearch()">
  <mat-icon>search</mat-icon>
</button>
```

```

onSearch () {
  if (!this.isValidSearch()) {
    return;
  }
  this.loading = true;
  this.accountId = null;
  this.formTitle = 'Transactions';
  this.transactionService
    .getTransactions(this.transactionsPerPage, 1, this.accountId, this.filterOption, this.filterString, this.dtp1.value, this.dtp2.value);
  this.subs = this.transactionService.getListUpdateListener()
    .subscribe((transactionData: {transactionData: TransactionModel[], transactionCount: number}) => {
      this.loading = false;
      this.currPage = 1;
      this.transactionData = transactionData.transactionData;
      this.totalTransactions = transactionData.transactionCount;
    });
}

isValidSearch () {
  let err = false;
  if (this.filterOption === 'amount') {
    this.openSnackBar('Please enter numeric value.', null);
    return false;
  }
  if (this.dtp1.enabled === true) {
    if (this.dtp1.value === null) {
      this.dtp1.setErrors({'incorrect': true});
      this.dtp1.markAsTouched();
      err = true;
    }
  }
  if (this.dtp2.value === null) {
    this.dtp2.setErrors({'incorrect': true});
    this.dtp2.markAsTouched();
    err = true;
  }
  if (err) {
    this.openSnackBar('Please select date range.', null);
    return false;
  }
  if (this.dtp1.value > this.dtp2.value) {
    this.openSnackBar('Error: From date is greater than until date.', null);
    return false;
  }
}
return true;
}

```


Backend

```
router.get("/", (req, res, next) => {
  const pageSize = +req.query.pageSize;
  const currPage = +req.query.currpage;
  var filter = req.query.filter;
  var value = req.query.value;
  var from = req.query.from;
  var until = req.query.until;
  let fetchedTransactions;
  let arr = [];
  var queryObj = {};
  function isValidDate(d) {
    if (Object.prototype.toString.call(d) === "[object Date]") {
      if (isNaN(d.getTime())) {
        return false;
      } else {
        return true;
      }
    } else {
      return false;
    }
  }
  console.log(filter);
  console.log(value);
  if (filter && value) {
    if (filter === "all" && !isNaN(value)) {
      queryObj = {
        $or: [
          { accId: new RegExp(value, "i") },
          { desc: new RegExp(value, "i") },
          { entered: new RegExp(value, "i") },
          { donation: new RegExp(value, "i") }
        ]
      };
    } else if (filter === "all" && !isNaN(value)) {
      queryObj = {
        $or: [
          { accId: new RegExp(value, "i") },
          { amount: value },
          { desc: new RegExp(value, "i") },
          { entered: new RegExp(value, "i") },
          { donation: new RegExp(value, "i") }
        ]
      };
    } else if (filter === "account") {
      queryObj = {
        $or: [{ accId: new RegExp(value, "i") }]
      };
    } else if (filter === "amount" && !isNaN(value)) {
      queryObj = {
        $or: [{ amount: value }]
      };
    } else if (filter === "desc") {
      queryObj = {
        $or: [{ desc: new RegExp(value, "i") }]
      };
    } else if (filter === "entered") {
      queryObj = {
        $or: [{ entered: new RegExp(value, "i") }]
      };
    } else if (filter === "donation") {
      queryObj = {
        $or: [{ donation: new RegExp(value, "i") }]
      };
    }
  }
  from = new Date (req.query.from);
  until = new Date (req.query.until);
  if (isValidDate(from) && isValidDate(until)) {
    arr = [{ date: { $gte: from } }, { date: { $lte: until } }];
    queryObj["$and"] = arr;
  }
  var query = TransactionSchema.find(queryObj);
  if (req.query.accId) {
    query = TransactionSchema.find({
      accId: req.query.accId
    });
  }
  if (pageSize && currPage) {
    query
      .sort("-date")
      .skip(pageSize * (currPage - 1))
      .limit(pageSize);
  }
  query
    .then(documents => {
      fetchedTransactions = documents;
      if (req.query.accId) {
        return TransactionSchema.countDocuments({
          accId: req.query.accId
        });
      } else {
        return TransactionSchema.countDocuments(queryObj);
      }
    })
    .then(count => {
      res.status(200).json({
        message: "Transactions fetched successfully!",
        transactions: fetchedTransactions,
        transactionCount: count
      });
    });
});
```

GUI

Transactions

Filter by	ter	Date Range	Choose from date	Choose until date	Q
All					
Account ID		Amount	Description	Entered by	
Amount		100	515	USER	
Description		-100	11	USER	
Entered By		121	2121	USER	
Donation					

Transactions

Filter by	Filter	Date Range	Choose from date	Choose until date	Q
All		8/1/2018	9/15/2018		
Account ID	Amount	Description	Entered by	Date	Donation
1231231 13 2123	10000	123213123	USER	2018-09-14T14:46:15.733Z	No
1111	-2000	qweqeq	USER	2018-09-14T07:02:28.025Z	No
1231231 13 2123	1111	qweweqwe	USER	2018-09-14T06:56:34.399Z	No
1111	123123123	323123 1 223	USER	2018-09-14T06:24:37.773Z	Yes
1111	1000	bh	USER	2018-09-14T06:22:18.838Z	Yes
Items per page: 15					1 - 5 of 5

Transactions

Filter by	Filter	Date Range	Choose from date	Choose until date	Q
Account ID	pro	8/1/2018	9/28/2018		
Account ID	Amount	Description	Entered by	Date	Donation
PROJ_Y	100	515	USER	2018-09-25T15:55:56.664Z	Yes
Items per page: 15					1 - 1 of 1

- Testing

Test Case	Search All Attributes
Expected Behavior	Display rows containing filter string
Test Steps	Select filter by field value as 'All' transaction page Enter Filter string Press search button
Test Status	Passed

Test Case	Search by Account ID
Expected Behavior	Display rows containing filter string as Account ID
Test Steps	Select filter by field value as 'Account ID' transaction page Enter Filter string

	Press search button
Test Status	Passed

Above Test Case done for all available Filters and all filters passed the test.

Test Case	Search by Date Range
Expected Behavior	Display rows entered between selected date range
Test Steps	Select filter by field value as 'All' Slide the Date Range Control. Enter From date Enter Until Date Press search button
Test Status	Passed

Test Case	Search by Date Range invalid
Expected Behavior	Display Error message- Error: From date is greater than until date.
Test Steps	Select filter by field value as 'All' Slide the Date Range Control. Enter From date Enter Until Date less than From date Press search button
Test Status	Passed

Commit IDs: [c18783a](#), [0c725fc](#), [1e3ff3f](#), [0a19f1d](#)

3.2 Detailed Account Page Search & Filtering Method Calls

Estimate Time: 2 Hours

Actual Time: 2 Hours

Actual Time (this sprint): 2 Hours

Description

This task is to enable filtering to transaction table.

```

<mat-form-field>
  <mat-select [(ngModel)]="filterOption" placeholder="Filter by">
    <mat-option value="all">All</mat-option>
    <mat-option value="account">Account ID</mat-option>
    <mat-option value="desc">Description</mat-option>
    <mat-option value="status">Status</mat-option>
  </mat-select>
</mat-form-field>
<mat-form-field>
  <input matInput [(ngModel)]="filterString" placeholder="Filter">
</mat-form-field>
<button mat-icon-button (click)="onSearch()">
  <mat-icon>search</mat-icon>
</button>
<mat-spinner *ngIf="loading"></mat-spinner>

```

Front end code

```

onSearch () {
  this.loading = true;
  this.accountService.getAccounts(this.accountsPerPage, 1, this.filterOption, this.filterString);
  this.subs = this.accountService.getListUpdateListener()
    .subscribe((accountData: {accountData: AccountModel[], accountCount: number}) => {
      this.loading = false;
      this.currPage = 1;
      this.accountData = accountData.accountData;
      this.totalAccounts = accountData.accountCount;
    });
}

```

Backend code

```
router.get("/", (req, res, next) => {
  const pageSize = +req.query.pageSize;
  const currPage = +req.query.currpage;
  var filter = req.query.filter;
  var value = req.query.value;
  var queryObj = {};
  console.log(filter);
  console.log(value);
  if (filter && value) {
    if (filter==='all'){
      queryObj = {
        $or:[{accId: new RegExp(value, "i")},{desc: new RegExp(value, "i")},{status: new RegExp(value, "i")}]
      }
    } else if (filter==='account'){
      queryObj = {
        $or:[{accId: new RegExp(value, "i")}]
      }
    } else if (filter==='desc'){
      queryObj = {
        $or:[{desc: new RegExp(value, "i")}]
      }
    } else if (filter==='status'){
      queryObj = {
        $or:[{status: new RegExp(value, "i")}]
      }
    }
  }
}

const query = AccountSchema.find(queryObj);
let fetchedAccounts;

if (pageSize && currPage) {
  query
    .sort("accId")
    .skip(pageSize * (currPage - 1))
    .limit(pageSize);
}

query
  .then(documents => {
    fetchedAccounts = documents;
    return AccountSchema.countDocuments(queryObj);
  })
  .then(count => {
    res.status(200).json({
      message: "Accounts fetched successfully!",
      accounts: fetchedAccounts,
      accountCount: count
    });
  });
});
```

GUI

Accounts

Accounts

Filter by
All ▼ Filter Q

Account ID	Description	Status	Create new			
1111	123123	Open	Add Entry	View Entries	Edit	Delete
1231231 13 2123	3 13 213 123123 12	Open	Add Entry	View Entries	Edit	Delete
1233	weew	Open	Add Entry	View Entries	Edit	Delete
43341	123123123131c	Close		View Entries	Edit	Delete
5646rtyh	fghfghfgh	Close		View Entries	Edit	Delete
7567n	th	Open	Add Entry	View Entries	Edit	Delete
PROJ_Y	Project Y account	Open	Add Entry	View Entries	Edit	Delete
PROJ_Z	Project Z account	Open	Add Entry	View Entries	Edit	Delete
asfaf	fasfa	Open	Add Entry	View Entries	Edit	Delete
b1	1	Open	Add Entry	View Entries	Edit	Delete
dfgsf gsf g fs gsf g s	g sfgsf g s g s	Open	Add Entry	View Entries	Edit	Delete
dsfsdfs sd sdf	sdf sdfsd f	Open	Add Entry	View Entries	Edit	Delete
fgdfgdfg	gdfgdfgd	Open	Add Entry	View Entries	Edit	Delete
fsdf dfsd ff sdf sdfsd fsdfsdf	fd fsdfsd ffs	Open	Add Entry	View Entries	Edit	Delete
gdfg dgd gdfg dfg	g g fg dfgdfg	Open	Add Entry	View Entries	Edit	Delete

Items per page: 15 1 - 15 of 24 < >

Accounts

Filter by
Account ID ▼ Filter pro Q

Account ID	Description	Status	Create new			
PROJ_Y	Project Y account	Open	Add Entry	View Entries	Edit	Delete
PROJ_Z	Project Z account	Open	Add Entry	View Entries	Edit	Delete

Items per page: 15 1 - 2 of 2 < >

• Testing

Test Case	Search All Attributes
Expected Behavior	Display rows containing filter string
Test Steps	Select filter by field value as 'All' Accounts page Enter Filter string Press search button
Test Status	Passed

Test Case	Search by Account ID
Expected Behavior	Display rows containing filter string as Account ID
Test Steps	Select filter by field value as 'Account ID' Accounts page Enter Filter string Press search button
Test Status	Passed

Above Test Case done for Description and Status Filters and all filters passed the test.

Commit IDs: [c18783a](#), [0c725fc](#), [0a19f1d](#)

3.3 Task 3 Account Report Page GUI Designing

Estimate Time: 2 Hours

Actual Time: 2 Hours

Actual Time (this sprint): 2 Hours


Description


Transaction report

From: 01/01/2018 Until: 31/01/2018

▼ Account	▼ Description	▼ Date	▼ Entered By	▼ Donation	▼ Amount
#####	#	#	#	#	#
#####	#	#	#	#	#
#####	#	#	#	#	#
#####	#	#	#	#	#
#####	#	#	#	#	#
#####	#	#	#	#	#
#####	#	#	#	#	#

Transaction Report

From Date: 

To Date: 

Donation: ▼

Account Transaction Report

From: 01/01/2018

Until: 31/01/2018

Account: Project Z

Description: Project Z Desc

Status: Open

▼ Description	▼ Date	▼ Entered By	▼ Donation	▼ Amount
#####	#	#	#	#
#####	#	#	#	#
#####	#	#	#	#
#####	#	#	#	#
#####	#	#	#	#
#####	#	#	#	#
#####	#	#	#	#

Account: Project Y

Description: Project Y Desc

Status: Closed

▼ Description	▼ Date	▼ Entered By	▼ Donation	▼ Amount
#####	#	#	#	#
#####	#	#	#	#

Account Transaction Report


Account Selection:


From Account: ▼

To Account: ▼

Status: ▼

Transaction Selection:

From Date:  ▼

To Date:  ▼

Donation: ▼

Generate

Account Balance Report

Report Filter Details

▼ Account	▼ Description	▼ Status	▼ Balance
#####	#	#	#
#####	#	#	#
#####	#	#	#
#####	#	#	#
#####	#	#	#
#####	#	#	#
#####	#	#	#

Account Balance Report

From Account:

 ▼

To Account:

 ▼

Status:

 ▼

Generate

Commit ID: [c9f2477](#)

3.9 Task 9 Sprint 5 Planning

Estimate Time:

1 Hours

Actual Time: 1 Hours

Actual Time (this sprint): 1 Hours

Description

This task is to plan upcoming sprint.

Please find the sprint 5 planning meeting minutes from the following link.

Bitbucket Link:

https://bitbucket.org/Computing_Projects_SLIIT/2018_sd07/src/master/Documents/Sprint%20Documents/Sprint%205/Sprint%205%20Planning%20Minutes.pdf

3.10 Task 10 Discuss design issues Sprint 4

Estimate Time: 1 Hours

Actual Time: 1 Hours

Actual Time (this sprint): 1 Hours

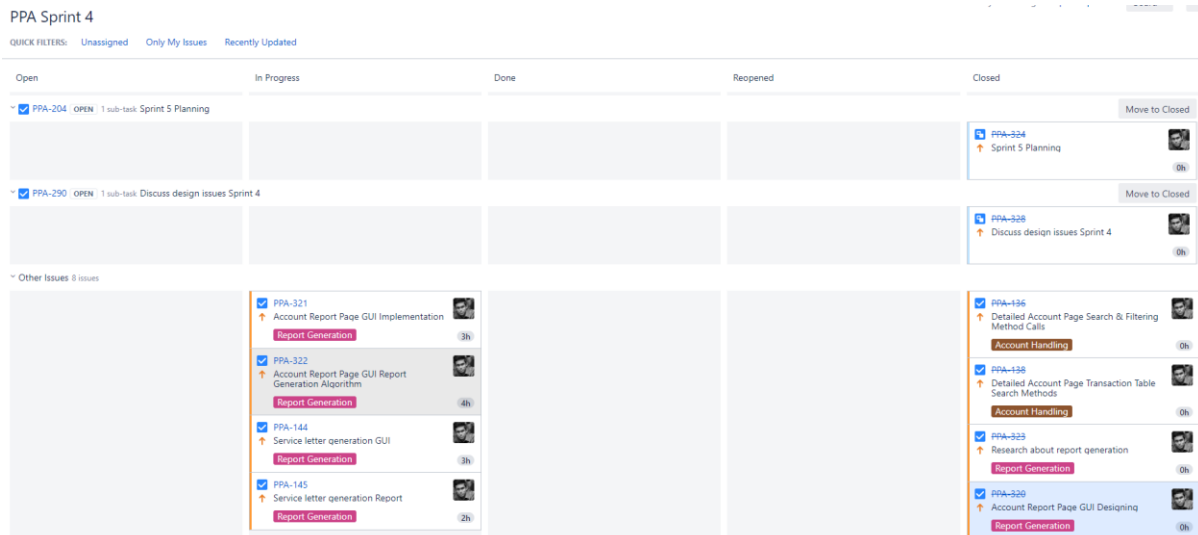
Description

This task is to discuss and solve technical and design issues.

4. Development Methodology

4.1 Minutes

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



We had 6 standup meetings within this sprint and the minutes of those meetings can be found from the following link.

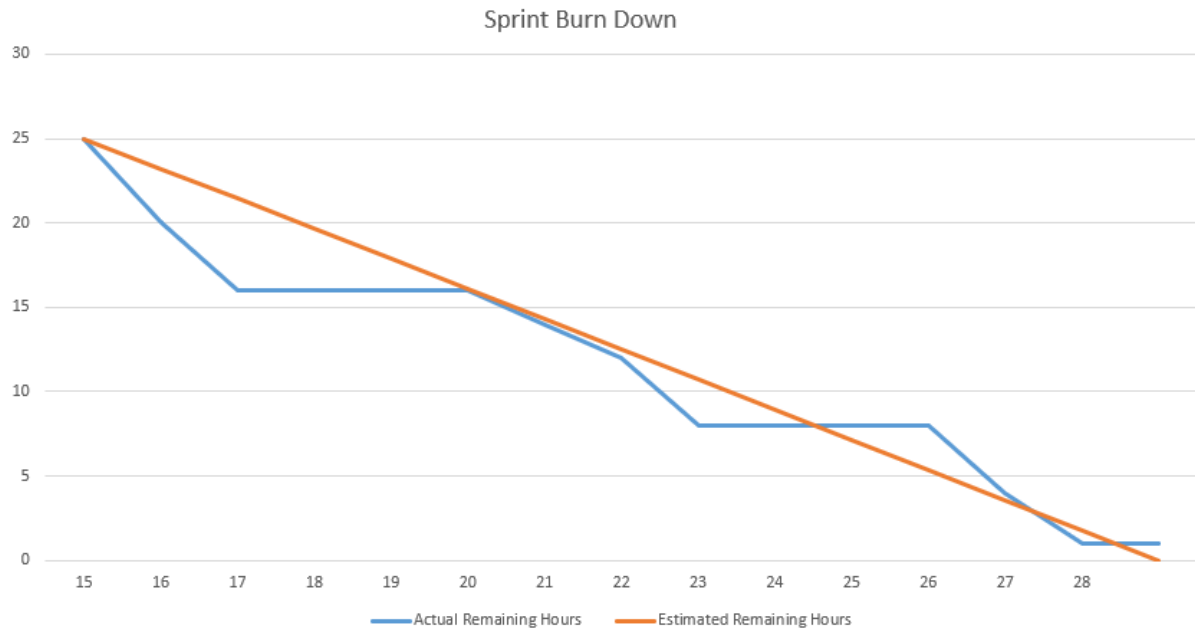
Standup Meeting Minutes:

https://bitbucket.org/Computing_Projects_SLIIT/2018_sd07/src/master/Documents/Standup%20Meeting%20Minutes/Sprint%204/Standup%20Meeting%20-%20Sprint%204.pdf

4.2 Burndown Chart

Estimate Time: 25 Hours

Actual Time: 24 Hours



4.3 Sprint Retrospective

Note: This is a continuing task for each sprint.

Estimate Time: 0.5 Hours

Actual Time: 0.5 Hours

Actual Time (this sprint): 0.5 Hours

4.4 Task Summary

Estimate Time: 25 Hours

Actual Time: 24 Hours

Completed 10 Task during this Sprint.

4.5 Time Management

#	Task ID	Task	Hours Estimated	Actual
1	PPA-	Detailed Account Page Transaction Table Search Methods	3	3
2	PPA-	Detailed Account Page Search & Filtering Method Calls	2	2
3	PPA-	Account Report Page GUI Designing	2	2
4	PPA-	Account Report Page GUI Implementation	3	3
5	PPA-	Account Report Page GUI Report Generation Algorithm	4	4
6	PPA-	Research about report generation	4	3
7	PPA-	Service Letter Generation GUI Designing & Implementation	3	3
8	PPA-	Service letter generation Report Designing (PDF)	2	2
9	PPA-	Sprint 5 Planning	1	1
10	PPA-	Discuss design issues Sprint 4	1	1
	Total		25	24