PPA Membership Maintenance System

Sprint Report 2

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:	31/08/2018	Which assignment?	Sprint Report 02

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: 31/08/2018

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

Table of Contents

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 2: 18th August – 31st August

2.1 Allocated Tasks for the Sprint 2

#	Task ID	Task	Task Status	Hours Estimated
1	PPA-	Learn Angular	Completed	4
2	PPA-	Gallery Page Main GUI Designing	Completed	1
3	PPA-	Gallery Page Main GUI Implementation	Completed	3
4	PPA-	Database Deigning	Completed	1
5	PPA-	Gallery Page Insert Methods	Completed	2
6	PPA-	Gallery Page Delete Methods	Completed	2
7	PPA-	Gallery Page Validations	Completed	2
8	PPA-	Main Accounts Page Designing & Implementation	Incomplete	3
9	PPA-	All Accounts Page Designing & Implementation	Incomplete	2
10	PPA-	Sprint 3 Planning	Completed	1
11	PPA-	Discuss design issues Sprint 2	Completed	1
			Total Hours	**22**

2.2 Planned Tasks for the Sprint 3

#	Task ID	Task	Estimation (Hours)
1	PPA-	Detailed Account Page Designing	1
2	PPA-	Detailed Account Page Implementation	4
3	PPA-	Detailed Account Page - Database Design	1
4	PPA-	Detailed Account Page - Insert Method Calls	2
5	PPA-	Detailed Account Page Update Method Calls	2
6	PPA-	Detailed Account Page Delete Method Calls	2
7	PPA-	Detailed Account Page Search & Filtering Method Calls	4
8	PPA-	Detailed Account Page Validation Methods	2
9	PPA-	Detailed Account Page Transaction Table Insert Methods	2
10	PPA-	Detailed Account Page Transaction Table Update Methods	3
11	PPA-	Sprint 4 Planning	1
12	PPA-	Discuss design issues Sprint 3	1
		Total Hours	**25**

2.3 Difficulties

3. Task Break Down

3.1 Task 1 Learn Angular

Estimate Time: 4 Hours

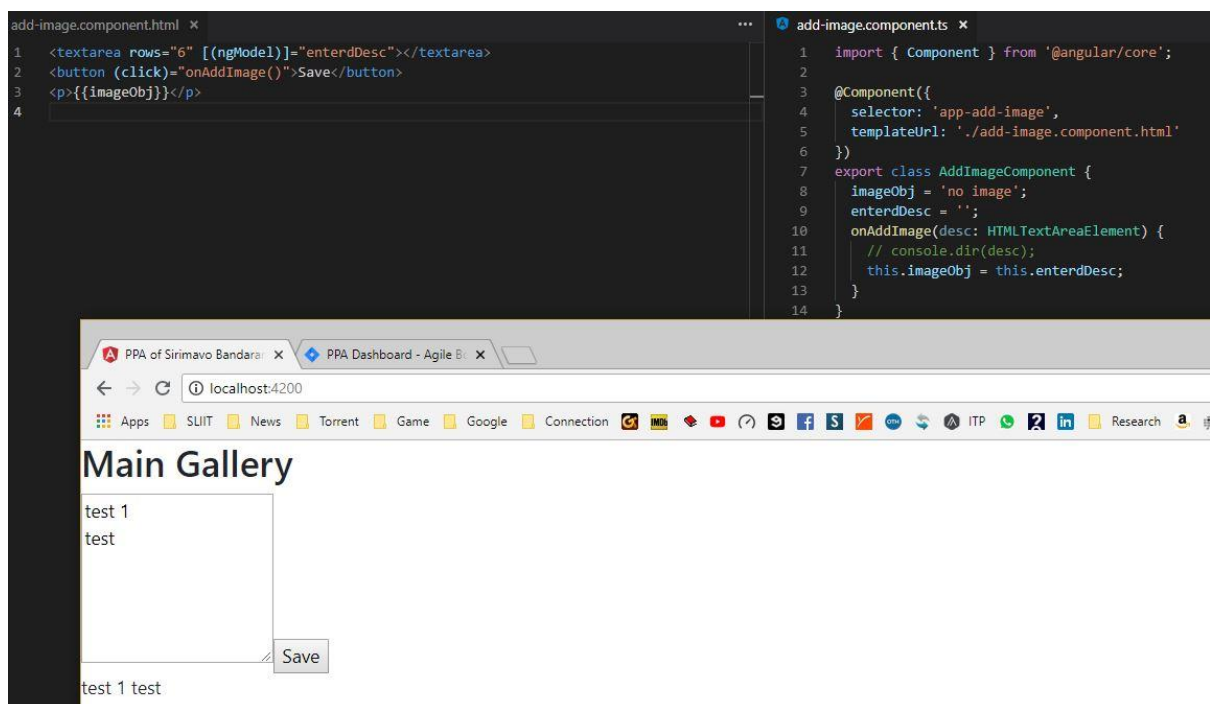Actual Time: 4 Hours

Actual Time (this sprint):   4 Hours

Description

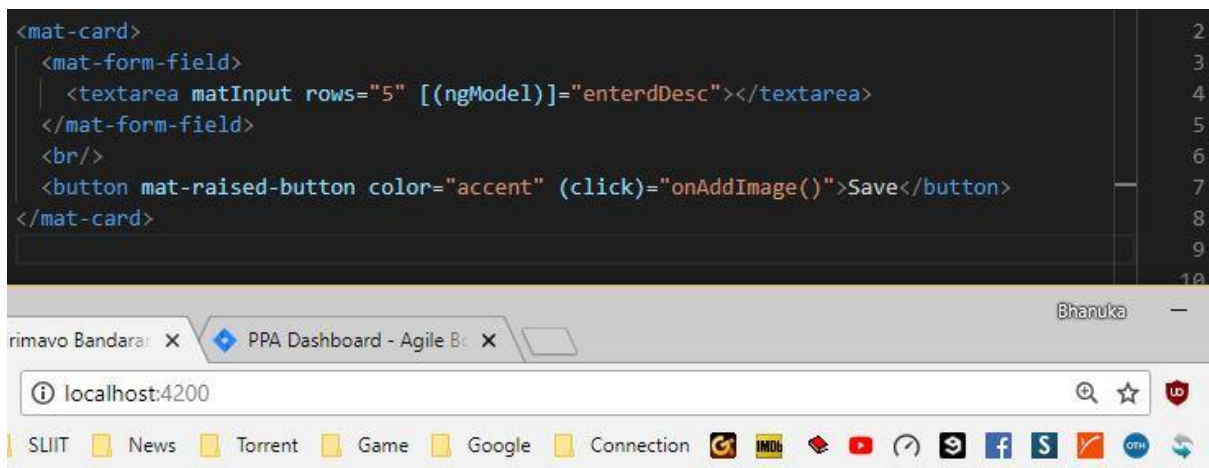This is task for learning Angular JS by following Udemi course (https://www.udemy.com/angular-2-and-nodejs-the-practical-guide/).

Below are some codes that I typed as I learn,
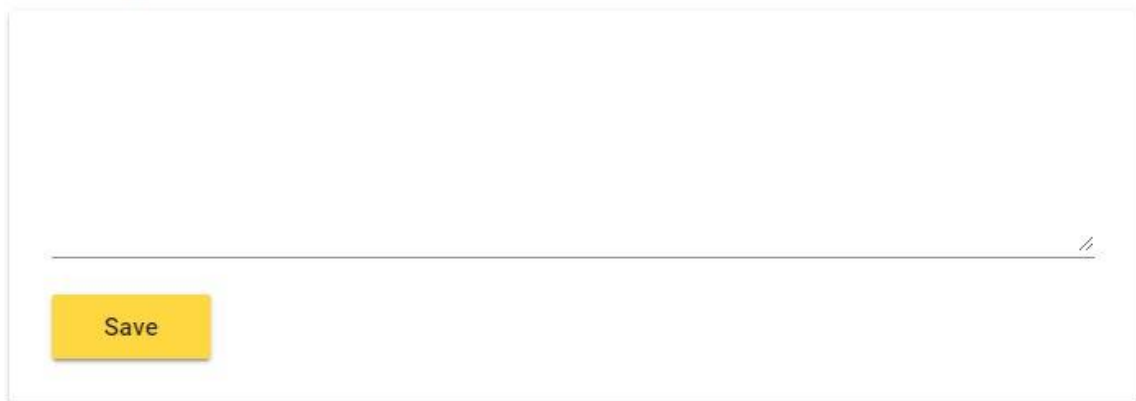
- Implementing two-way binding

- Use of angular material

```
<mat-card>
  <mat-form-field>
    <textarea matInput rows="5" [(ngModel)]="enterdDesc"></textarea>
  </mat-form-field>
  <br/>
  <button mat-raised-button color="accent" (click)="onAddImage()">Save</button>
</mat-card>
```

Gallery

[textarea]

[Save]

- string interpolation, button events and property binding

```
<textarea rows="6" [value]="imageObj" #imageDesc></textarea>
<button (click)="onAddImage(imageDesc)">Save</button>
<p>{{imageObj}}</p>
```

```
import { Component } from '@angular/core';

@Component({
  selector: 'app-add-image',
  templateUrl: './add-image.component.html'
})
export class AddImageComponent {
  imageObj = 'no image';
  onAddImage(desc: HTMLTextAreaElement) {
    // console.dir(desc);
    this.imageObj = desc.value;
  }
}
```

Main Gallery

no imagesss

[Save]

no imagesss
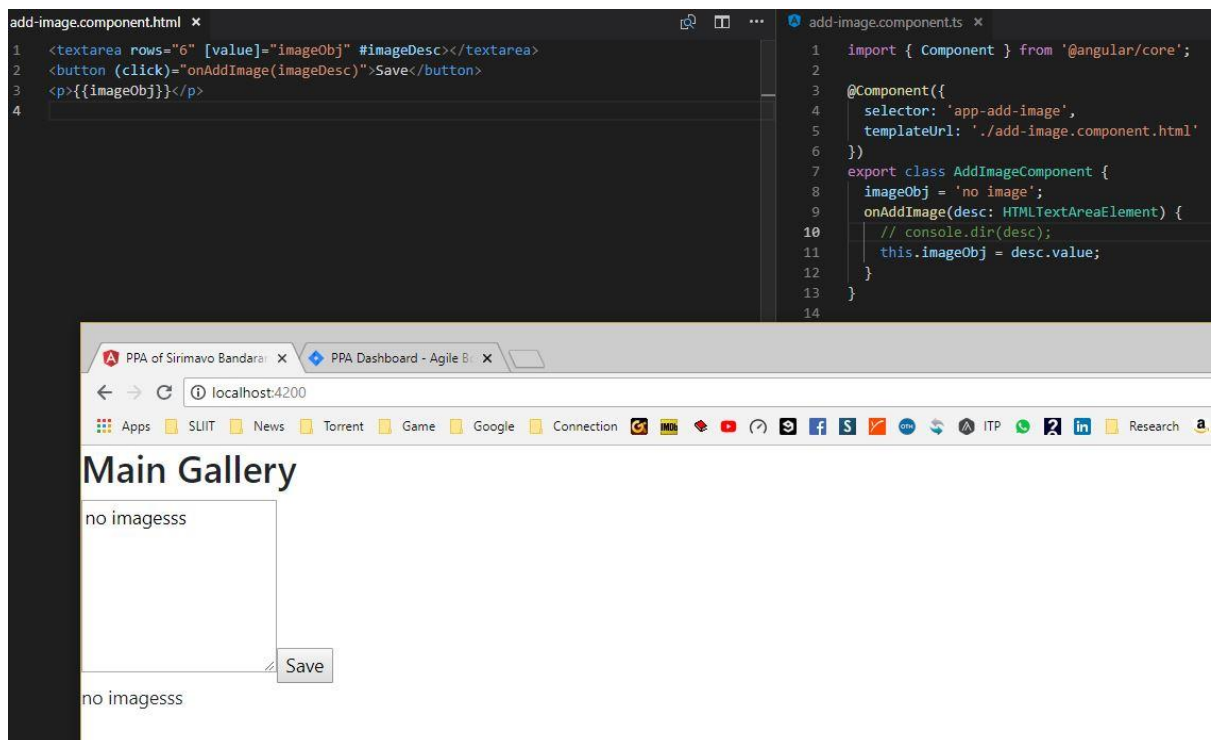
3.2 Task 2 Gallery Page Main GUI Designing

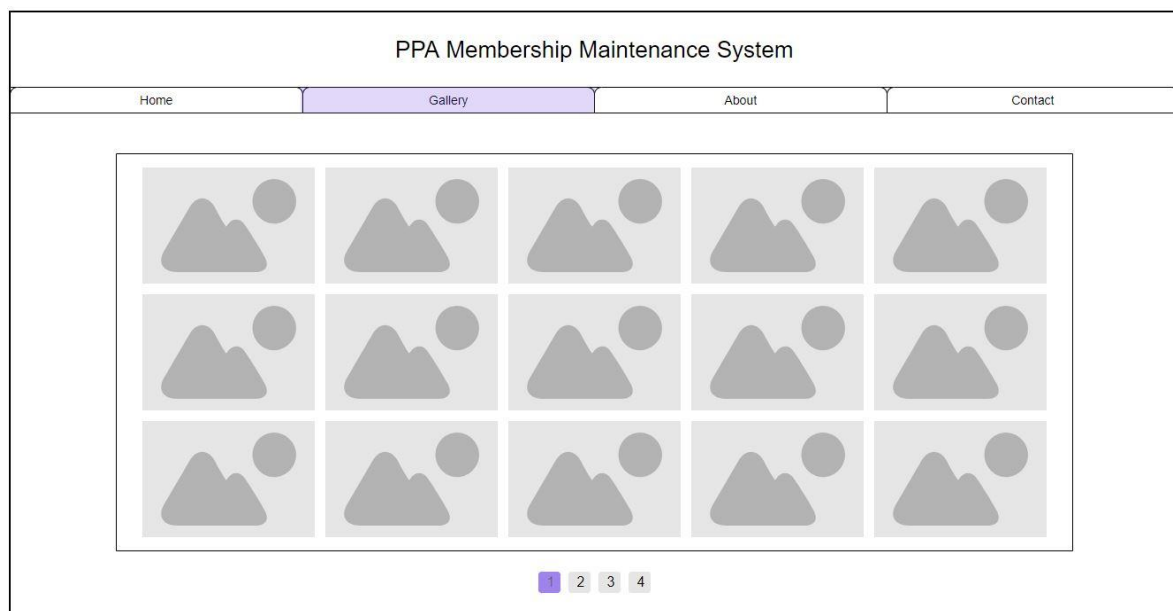Estimate Time: 1 Hours

Actual Time: 1 Hours

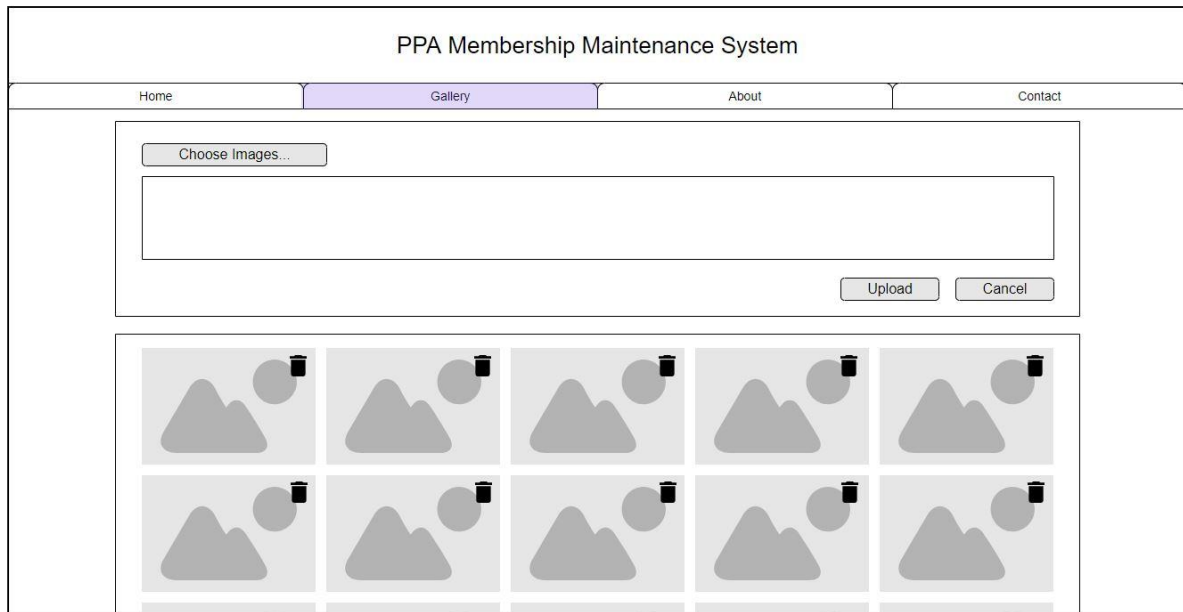Actual Time (this sprint):   1 Hours

Description

From this task I designed draft GUIs for Main Gallery page. My work commit link is given below.

https://bitbucket.org/Computing_Projects_SLIIT/2018_sd07/commits/6c527a52c15fd54e0d881a302d205c9f999f2f2b
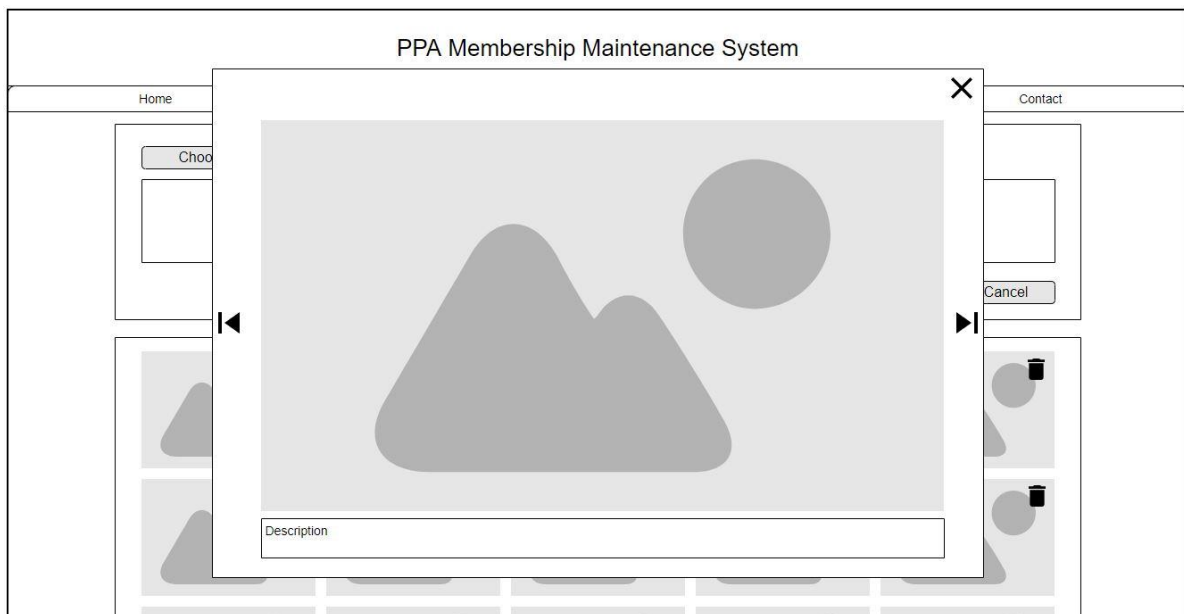

Normal user view of the gallery

Admin view of the gallery

When developing I added an extra field when inserting image to save the title.

Maximized view of a Image

3.3 Task 3 Gallery Page Main GUI Implementation

Estimate Time: 3 Hours

Actual Time: 4 Hours

Actual Time (this sprint):   4 Hours

Description

These pages were design as components that is to be loaded inside a parent form that has general navigations, It is the best approach when developing an angular web application.

Admin view of the gallery page

Normal user view of the gallery page

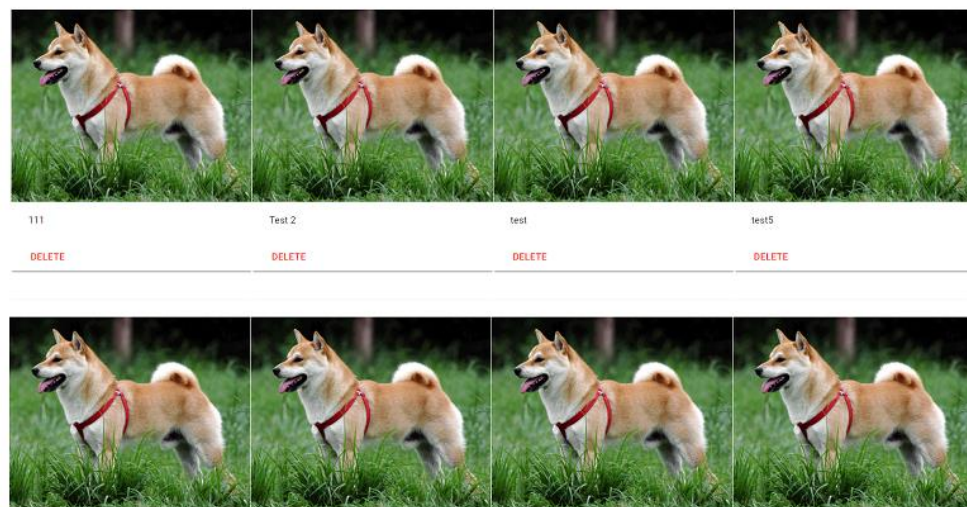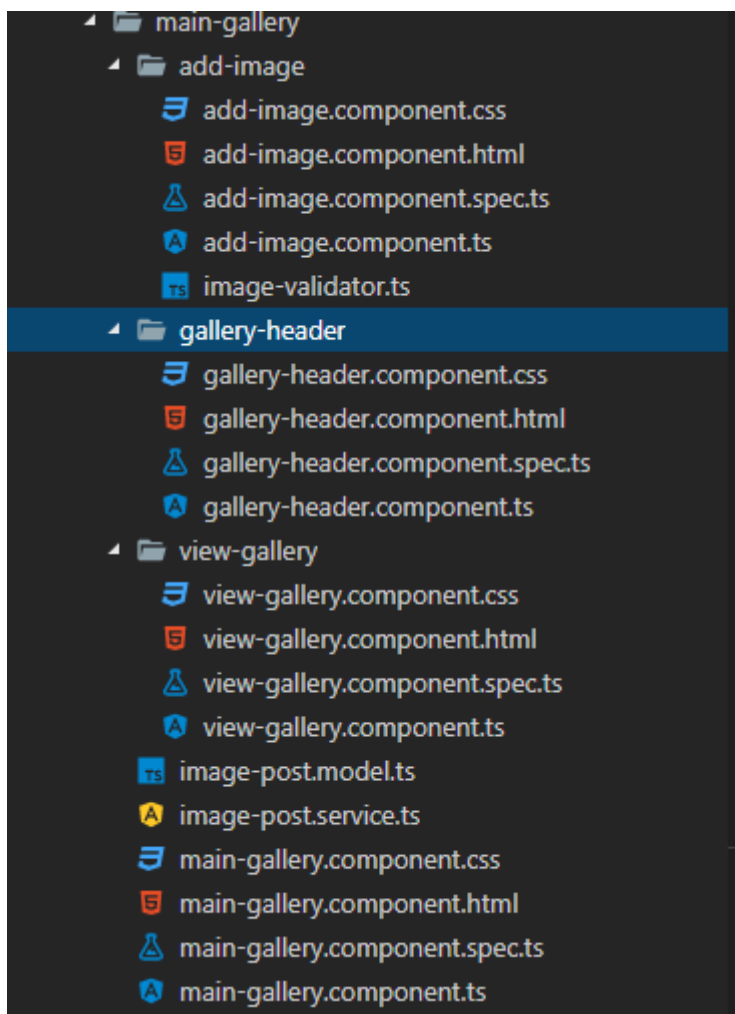*4 components used when developing gallery page.*

```
<mat-card>
    <form [formGroup]="form" (submit)="onAddImage()">
        <div>
            <button mat-stroked-button type="button" (click)="filePicker.click()">Pick Image</button>
            <input type="file" #filePicker (change)="onImagePicked($event)">
        </div>
        <div class="image-preview" *ngIf="imagePreview !== '' && imagePreview && form.get('image').valid">
            <img [src]="imagePreview" [alt]="form.value.title">
        </div>
        <mat-form-field>
            <input
                matInput
                type="text"
                formControlName="title"
                placeholder="Image Title">
            <mat-error *ngIf="form.get('title').invalid">Please enter a title.</mat-error>
        </mat-form-field>
        <mat-form-field>
            <textarea
                matInput
                rows="4"
                formControlName="desc"
                placeholder="Description"></textarea>
            <mat-error *ngIf="form.get('desc').invalid">Please enter a post title.</mat-error>
        </mat-form-field>
        <button
            mat-raised-button
            color="accent"
            type="submit">Save Post</button>
    </form>
</mat-card>
```

Insert image GUI code

```
<div class="main-view-div">
  <mat-grid-list cols="4" rowHeight='500px' *ngIf="images.length > 0">
    <mat-grid-tile *ngFor="let imgcard of images">
      <mat-card class="photo-card">
        <mat-card-header>
        </mat-card-header>
        <img mat-card-image src="https://material.angular.io/assets/img/examples/shiba2.jpg" [alt]= "imgcard.title">
        <mat-card-content>
          <p> {{ imgcard.desc }} </p>
        </mat-card-content>
        <mat-card-actions>
          <button mat-button color="warn" (click)="onDelete(imgcard.id)">DELETE</button>
        </mat-card-actions>
      </mat-card>
    </mat-grid-tile>
  </mat-grid-list>
  <p class="no-image-info mat-body-1" *ngIf="images.length < 1">No Imags</p>
</div>
```

View image GUI code

3.4 Task 4 Database Deigning (Main Gallery)

Estimate Time: 1 Hours

Actual Time: 0.5 Hours

Actual Time (this sprint): 0.5 Hours

Description

From this task I have decided the attributes that need to implement a gallery in expected manner.

I have decided to use following attributes,

- Uploaded date
- Image Path
- Image Description
- Image Title
- Id (default mongo DB ID)

Reason of using image path instead of storing the image is that its efficient to store the image as it is in the server and giving the access to that directory to the client. Converting images when storing and retrieving will take more time.

3.5 Task 5 Gallery Page Insert Methods

Estimate Time: 2 Hours

Actual Time: 2 Hours

Actual Time (this sprint): 2 Hours

Description

3.6 Task 6 Gallery Page Delete Methods

Estimate Time: 2 Hours

Actual Time: 2 Hours

Actual Time (this sprint): 2 Hours

Description

Below is the back end code of the insert logic.

```
const storage = multer.diskStorage({
  destination: (req, file, cb) => {
    const isValid = MIME_TYPE_MAP[file.mimetype];
    let error = new Error("Invalid type!");
    if (isValid) {
      error = null;
    }
    cb(error, "server/images");
  },
  filename: (req, file, cb) => {
    const name = file.originalname.toLocaleLowerCase().split(' ').join('-');
    const ext = MIME_TYPE_MAP[file.mimetype];
    cb(null, name + "-" + Date.now() + "." + ext);
  }
});

router.post("/", multer({storage: storage}).single("image"), (req, res, next) => {
  const url = req.protocol + '://' + req.get("host");
  const post = new GallerySchema({
    title: req.body.title,
    desc: req.body.desc,
    imagePath: url + "/images/" + req.file.filename
  });
  post.save().then(createdPost => {
    res.status(201).json({
      message: "Image added successfully",
      imgObj: {
        ...createdPost,
        id: createdPost._id
      }
    });
  });
});
```

3.7 Task 7 Gallery Page Validations

Estimate Time: 2 Hours

Actual Time: 2 Hours

Actual Time (this sprint): 2 Hours

Description

Pick Image

Image Title

Please enter a title.

Description

Please enter a post title.

Save Post

```
ngOnInit() {
  this.form = new FormGroup({
    title: new FormControl(null, {
      validators: [Validators.required]
    }),
    desc: new FormControl(null, { validators: [Validators.required]}),
    image: new FormControl(null, {validators: [Validators.required], asyncValidators: [imageValidator]})
  });
}
}
```

Form validation code

```typescript
export const imageValidator = (
  control: AbstractControl
): Promise<{ [key: string]: any }> | Observable<{ [key: string]: any }> => {
  if (typeof(control.value) === 'string') {
    return of(null);
  }
  const file = control.value as File;
  const fileReader = new FileReader();
  const frObs = Observable.create(
    (observer: Observer<{ [key: string]: any }>) => {
      fileReader.addEventListener('loadend', () => {
        const arr = new Uint8Array(fileReader.result).subarray(0, 4);
        let header = '';
        let isValid = false;
        for (let i = 0; i < arr.length; i++) {
          header += arr[i].toString(16);
        }
        switch (header) {
          case '89504e47':
            isValid = true;
            break;
          case 'ffd8ffe0':
          case 'ffd8ffe1':
          case 'ffd8ffe2':
          case 'ffd8ffe3':
          case 'ffd8ffe8':
            isValid = true;
            break;
          default:
            isValid = false;
            break;
        }
        if (isValid) {
          observer.next(null);
        } else {
          observer.next({ invalidMimeType: true });
        }
        observer.complete();
      });
      fileReader.readAsArrayBuffer(file);
    }
  );
  return frObs;
```

Image type validation code

3.8 Task 8 Main Accounts Page Designing & Implementation

Estimate Time: 3 Hours

Actual Time: 3 Hours

Actual Time (this sprint):   3 Hours

Description

Draft Design ->
https://bitbucket.org/Computing_Projects_SLIIT/2018_sd07/changeset/353adf28f2f2411eb3d2a1f49ecba01e33989fb3

3.9 Task 9 All Accounts Page Designing & Implementation

Estimate Time: 2 Hours

Actual Time: 2 Hours

Actual Time (this sprint): 2 Hours

Description

All Account page is only containing a table that displays all accounts entered by the users.

3.10 Task 10 Sprint 3 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 3 planning meeting minutes from the following link.

Bitbucket Link:

https://bitbucket.org/Computing_Projects_SLIIT/2018_sd07/src/master/Documents/Sprint%20Documents/Sprint%202/Sprint%202%20Planning%20Minutes.pdf

3.11 Task 11 Discuss design issues Sprint 2

Estimate Time: 1 Hours

Actual Time: 2 Hours

Actual Time (this sprint): 2 Hours

Description

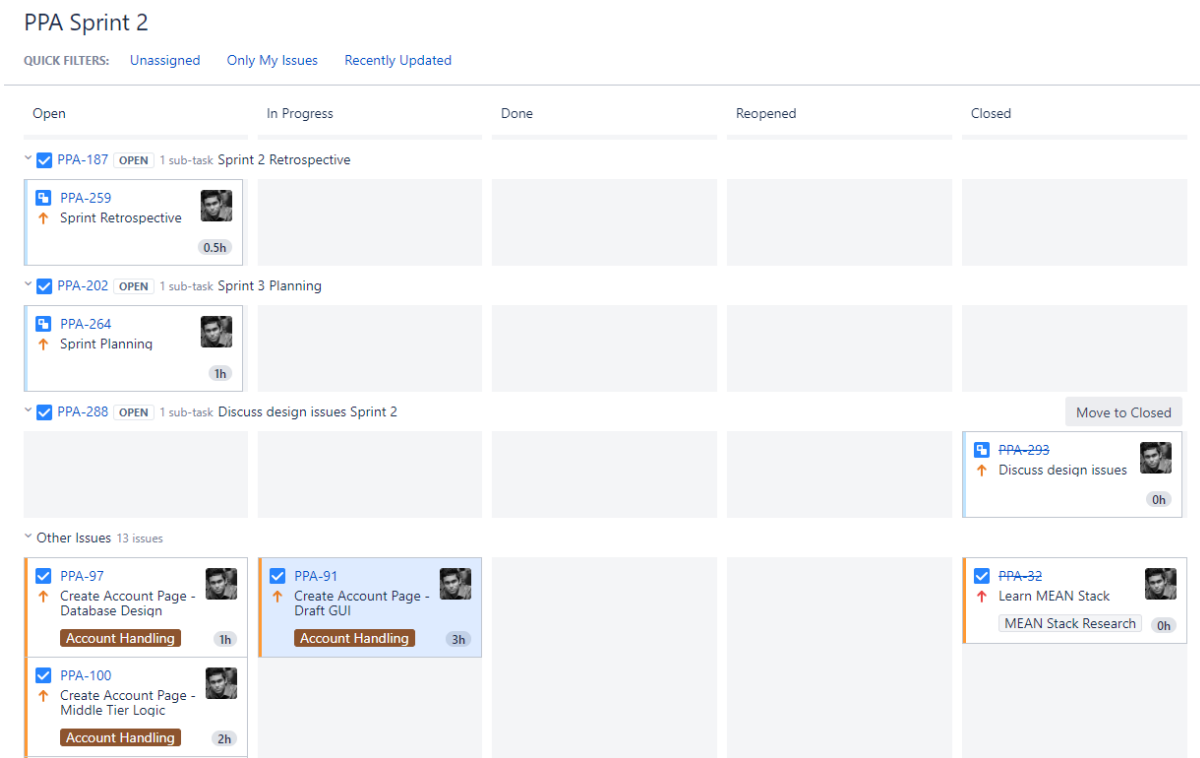This task is to discuss and solve technical and design issues.

Meeting minutes ->

https://bitbucket.org/Computing_Projects_SLIIT/2018_sd07/src/master/Documents/Design%20Meetings/Design%20Meeting%20-%20Sprint%202.docx

4. Development Methodology

4.1 Minutes

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


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%202/Standup%20Meeting%20-%20Sprint%202.pdf

4.2 Burndown Chart

Estimate Time: 22 Hours

Actual Time: 23.5 Hours

Sprint Burn Down

Legend: Actual Remaining Hours, Estimated Remaining Hours
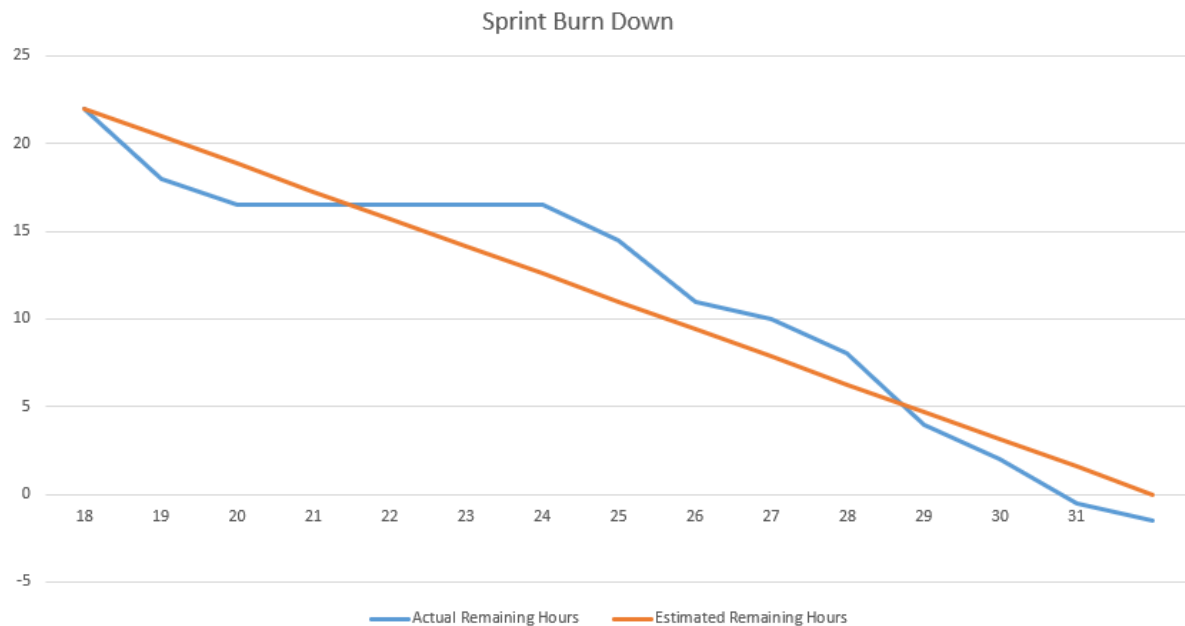
4.3   Sprint Retrospective

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

Description

In this sprint we started our developments. Since all of us have not a much exposure in MEAN stack domain, there were quite few learning stuff during the developments. We hope that the exponential growth of the learning curve will be come to a more flatten phase in the future sprints as we get familiar with the language and frameworks. As development goes on the heroku build which runs builds after every commit in the online repo was really helpful to detect the errors that were rose when merging. That was easy for us to fix the relevant issue, since that tool points out which commit cause to the error. Some of the initial tasks of the sprint were taken a bit additional time than allocated due to the learning stuff. Apart from that all the planned tasks were went well as expected. We had about 4 standup meetings time to time, to discuss the current status of the each ones allocated tasks. We always tried to keep the look and feel of the UIs by discussing and reviewing each other's work during these meetings. We also used a common flow of development

to achieve the tier based architecture which was decided in the design meeting. Jira tool was quite helpful to track each of us progress during the sprint. Its real time sprint burndown chart was came quite handy when estimating and planning the remaining workload of the sprint, in both perspective as a team as an individual. Please refer the following two snapshots of the burndown chart (in team perspective and individual perspective) which were taken at some point during the sprint 2 for more clearance.

4.4   Task Summary

Estimate Time: 22 Hours

Actual Time: 23.5 Hours

Completed 9 Task during this Sprint.

4.5   Time Management

#	Task ID	Task	Hours Estimated	Actual
1	PPA-	Learn Angular	4	4
2	PPA-	Gallery Page Main GUI Designing	1	1
3	PPA-	Gallery Page Main GUI Implementation	3	4
4	PPA-	Database Deigning	1	0.5
5	PPA-	Gallery Page Insert Methods	2	2
6	PPA-	Gallery Page Delete Methods	2	2
7	PPA-	Gallery Page Validations	2	2
8	PPA-	Main Accounts Page Designing & Implementation	3	3
9	PPA-	All Accounts Page Designing & Implementation	2	2
10	PPA-	Sprint 3 Planning	1	1
11	PPA-	Discuss design issues Sprint 2	1	2
	Total		**22**	**23.5**