***HSC Major project Stage 3***

**PROJECT Development Documentation**

**Student Name: Jacob Greenberg**

**Class: Software Design & Development**

**Teacher: Peter Cox**

**TABLE OF CONTENTS**

[System Design 3](#_Toc384908244)

[TASK 1 – PLANNING and WORKLOG 3](#_Toc384908245)

[TASK 2 – IMPLEMENTATION (Coding) 3](#_Toc384908246)

[Code Sample 1 3](#_Toc384908247)

[Code Sample 2 4](#_Toc384908248)

[Code Sample 3 4](#_Toc384908249)

[TASK 3 – TESTING 5](#_Toc384908250)

[Coding log 5](#_Toc384908251)

[Version control and backup 6](#_Toc384908252)

[Function Testing – 3 samples required 6](#_Toc384908253)

[Module list 7](#_Toc384908254)

[TASK 4 – USER DOCUMENTATION 8](#_Toc384908255)

[License agreement and copyright details 8](#_Toc384908256)

[Copyright license 8](#_Toc384908257)

[Contact details 8](#_Toc384908258)

[Installation guide 9](#_Toc384908259)

[TASK 5 – EXHIBITION 10](#_Toc384908260)

[TASK 6 – REFLECTION and PROGRAM STATUS REPORT 10](#_Toc384908261)

[Reflection You must Complete the personal reflection worksheet. 10](#_Toc384908262)

[Status report 10](#_Toc384908263)

# System Design

Attached is a copy of the system design from stage 2 of the project, including structure diagram and module list.

# TASK 2 – IMPLEMENTATION (Coding)

## Code Sample 1: HTML5 Chart Draw

Part of the program it comes and what it does:

This function is called, when the user visits the student dashboard. It is used to reliably initialise a HTML5 chart, which the student assessment data can then be plotted on. The chart provides a clean, user friendly way of displaying data.

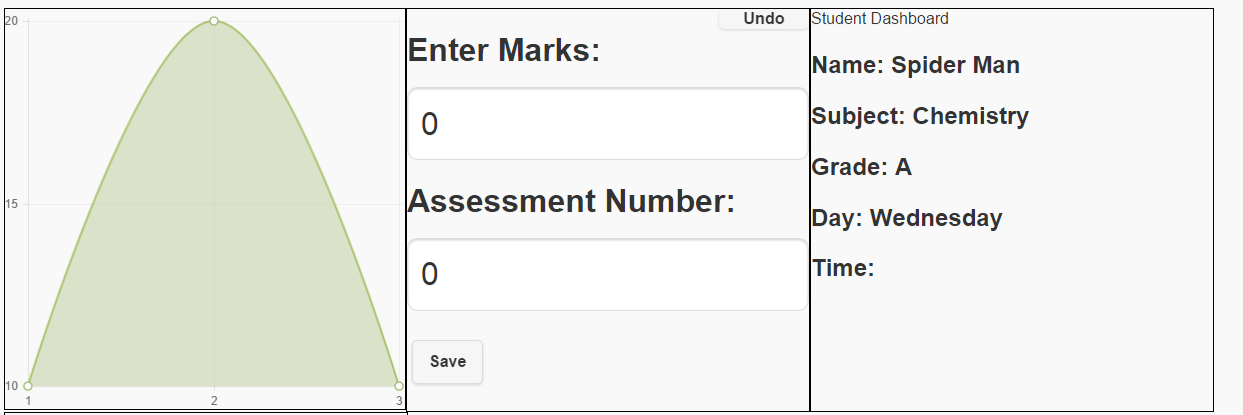
Code Example



Explanation of Code:

1. The code erases the previous canvas, to ensure their will be no overlap or clash when it is finally rendered by your browser.
2. The element is then recreated, and appended to the page.
3. The function which actually specifies the charts data is called.
4. Chart object is initialised.
5. A JSON format dataset is created, which specifies line colour, plotted point colour. These parameters are purely aesthetic.
6. In the final JSON object, the data is provided. It is inputted by the user, and stored in local storage. This pulls the data from local storage and graphs it.

**Picture of GUI**

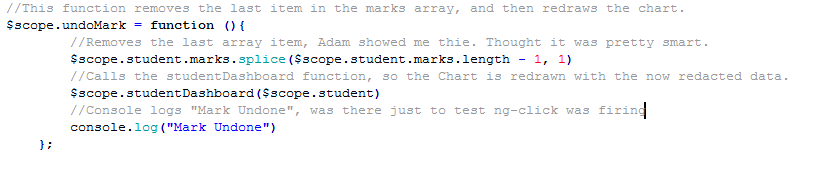


## Code Sample 2

Part of the program it comes and what it does.

On the student dashboard, if you want to undo the graphed mark input there is a button in the input section which allows you do to do so. When clicked, the inputted mark/assessment number is popped from local storage and the graph is redrawn to display what it now looks like.

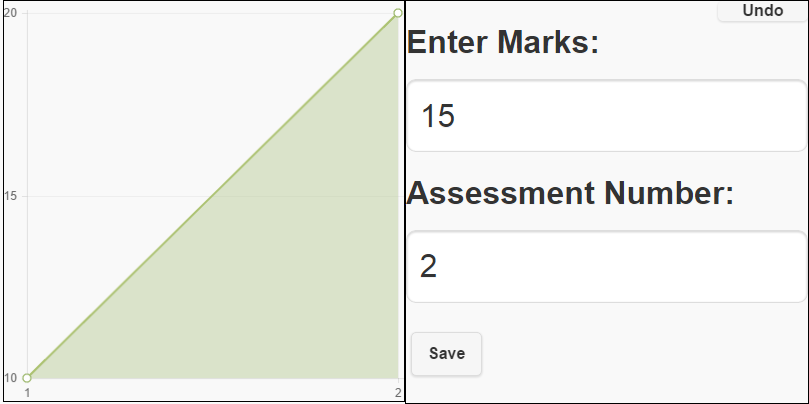
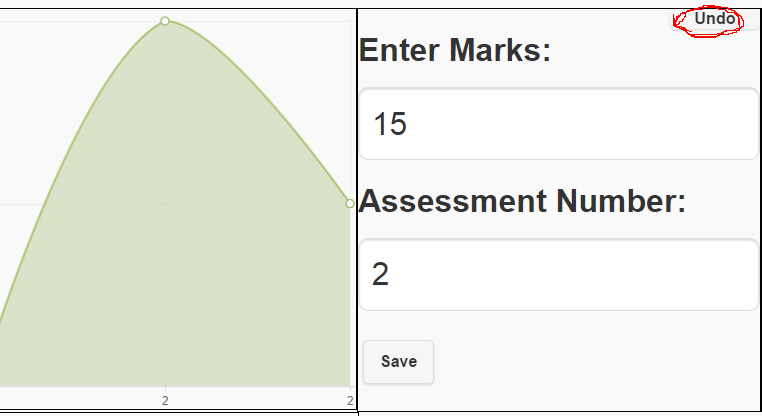
Code Example



Explanation of Code

1. Removes the last item of the array, Adam taught me how to do this and I think it’s really smart. It determines the last item of the array, by determining the array length and subtracting 1.
2. Calls the studentDashboard function, which effectively redraws the chart with the now redacted array of marks
3. The console logs as “mark undone”

Screenshot

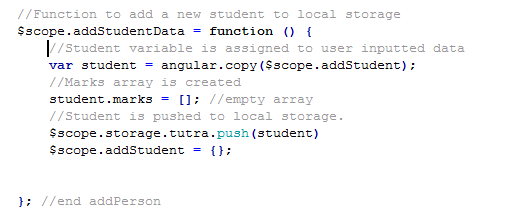


## Code Sample 3

Part of the program it comes and what it does.

When it comes to adding the student, this function is called after the user has inputted all the necessary data. It takes the user inputted data and pushes them to local storage.

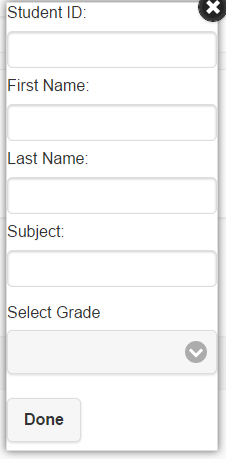
Code Example



Explanation of Code

1. Student variable is assigned to ng-model inputs
2. An empty array to hold the students marks is created
3. The student data, is now pushed to local storage.

Screenshot



# TASK 3 – TESTING

## Coding log

*The following is a log of modules worked on, recording when, what completed and features functions still to implement*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Date** | **Module/Function** | **Work done** | **Functions completed (% done)** | **Evidence** of the testing and debugging. **What** **Issues, problems, actions did you do** |
| *27/4* | *Got my prototype AngularJS ready* | *Used peters outline, to setup an angular controller and point it to my page.* | *100%* | ***Error****: Angular was not coming through e.g {{1+1}} not working* ***Error type****: Syntax*  ***Debug****: My angular controller was called “myApp” and the angular module in index.html was just called app.* ***Solution****: Renamed the angular module to MyApp.* |
| *27/4* | *Follow Peter’s local storage template.* | *Copied what peter did exactly.*  *Still don’t really understand how it’s used.* | *100%* | ***Error****: None* ***Error type****: None* ***Action****: None* ***Solution:****None* |
| 01/5 | *Adding students* | *Created a pop up iframe, which had a series of forms which the user can input student data into.* | *30%* | *Error: Couldn’t define the height and width of the iframe****.******Error Type:*** *Syntax*  ***Debug:*** *I ended the last CSS section with a semi colon, so the next was not being loaded.* ***Solution:*** *Added a semi colon.* |
| 08/5 | Adding students | Peter got the user info onto the scope, and pushed to local storage. Adam helped me append the user to a list. | 80% | There could have been some errors, but Peter and Adam kind of handed it to me done. |
| 10/5 | Adding students | Peter made it so, when a user is clicked on in the list, it links to a page which dynamically displays the data of that student. This is basically what I wanted, adding students is down. | 100% | Error: Angular was not working all of a sudden.  Error Type: Network  Debug: Fetching AngularJS remotely just wasn’t working for some reason.  Solution: Downloaded Angular, and ran it locally. |
| 18/5 | Graphing functionality | Using ChartJS as my graphing framework, I downloaded the JS files and followed the docs. I was just trying to create a dummy graph, following the docs. But it’s just not showing up. | 5% | **Error: Uncaught TypeError: Cannot read property 'getContext' of null**  **Error Type: Syntax?**  **Debug: No idea, Google isn’t yielding results**  **Solution: No idea** |
| 19/5 | Graphing functionality | Ok, got it working. JS wasn’t picking up myChart because it was taking too long to render. So when the script ran it didn’t actually exist yet. Fixed this, by calling the function onLoad. | 45% | **Error: Uncaught TypeError: Cannot read property 'getContext' of null**  **Error Type: Syntax?**  **Debug: The script was running, but the chart had not been loaded yet. So to the script, the chart didn’t exist.**  **Solution: Call the script onLoad.** |
| 20/5 | Day/Time, Marks input and storage. | I have put 2 input boxes x2 on the student dashboard. The 1st set stores the assessment number and mark, the 2nd set stores the time and day of the student’s lesson. They are both now storing to the database. | 100% | **Error: ng-click wasn’t firing**  **Error Type: Syntax**  **Debug: Buttons I had copy pasted, were using “ng-cick“**  **Solution: Changed “ng-cick” to “ng-click”** |
| 21/4 | User editable graph. | This…is going to be a big task, I’m probably going to need to enlist the help of Adam. I’ve spent like an hour now, just messing around with this. I can get the chart to update, but it’s not storing anything. | 30% | **Error: The Assessment number, which sits on the Y axis no matter what the user entered was coming up as “undefined”**  **Error Type: Probably syntax**  **Debug: The assessment number input, was for some reason not properly being put on the scope.**  **Solution: Peer review and debugging.** |
| 22/4 | Graph storage. | I’ve uploaded my code to GitHub and am going to let a 3rd party help and walk me through what it’s going to take to have the stored mark/assessment numbers plotted back on the graph every time you revisit the page. | 30% | **Error:N/A**  **Error Type: N/A**  **Debug: N/A**  **Solution: N/A** |
| 22/4 | Graph storage | The issue for starters is, I was storing the marks/assessment numbers in their own database instead of under the student object. This would make it difficult to tie specific marks to particular students, so Adam walked me through fixing this. The marks and assessment numbers also needed to be stored in an array format amongst other things so Adam did a major commit on GitHub and pushed the update to me. It is now working as it should. | 30%  100% | **Error: Wrong use of database**  **Error Type: Logic**  **Debug: The mark/assessment data needed to be died to the student object in as an array. Not in a database of its own.**  **Solution: Stored the assessment/mark data as an array under student.** |
| 22/5 | Delete student | I used Peters Angular demo to do this, pretty much just copied his code 1 for 1. Yielded pretty much the expected results, no major issues. I also added a delete confirmation dialogue. | 100% | **Error: For some reason, when deletion was confirmed it deleted the last item in the list not the one you selected.**  **Debug: Peters delete confirmation is called with the parameter (index), I was calling mine with $index.**  **Solution: Changed the parameter, and got expected results.** |
| 23/5 | Edit student | Used peters Angular demo to add the ability to edit the student, also just pretty much copied it 1 for one. I don’t really understand how the confirmation button works, but it’s unnecessary. I even think editing happening real time is a cool little feature. | 100% | **Error: N/A**  **Error Type: N/A**  **Debug: N/A**  **Solution: N/A** |
| 24/5 | Undo mark | Initially I was thinking, I could have like a whole other which displayed all the students’ marks/assessment numbers but I figured on a mobile device that could get annoying. The user wants as easy as experience as possible. So I added an undo button, which when clicked erases the last added mark/assessment number from the database and redraws the chart. Adam gave me a handy technique for removing the last item in the array. | 100% | **Error: ReferenceError: student is not defined**  **Error Type: Syntax**  **Debug: I was splicing from the array student.marks, but forget to add that it was on the scope.**  **Solution: student.marks, because $scope.student.marks** |

## Version control and backup

*The following is a record of the software versions and backups.*

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Date** | **Summary of working functions and features, or changes to previous versions and where saved** | **Known issues, problems, actions to do** |
| *0.1* |  | *Template is there from Section 2, but nothing is functional.* | *Database not yet set up* |
| *0.2* |  | *Added* |  |
| *0.3* |  | *Created database and tables, View page , part functional* |  |
|  |  |  |  |
|  |  |  |  |

## Function Testing – 3 samples required

**Add Student Module**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ***Brief description of function*** | | ***Explanation*** | | |
| *User wants to add a new student to the list and database.* | | *A tutor wants to add a new student to the Tutra app database, so he can track their progress.* | | |
| *Pre-condition:* | | *The database must be built, which it is done automatically with a hardcoded “Spider Man” DB template.* | | |
| *Page descriptor:* | | *“Students Page”* | | |
| *Screen shot* | |  | | |
| Input data | Expected output | | Actual output | Status |
| 1 | Jacob added to the list, with student ID in the database. | | Jacob added to the list, with student ID in the database. | Functional |
| “Jacob” | Jacob added to the list. | | Jacob added to the list. | Functional |
| “Greenberg” | Jacob Greenberg added to the list of students. | | Jacob Greenberg added to the list of students. | Functional |
| “Maths” | Below the student name, the subject: with in this case it being Maths. | | Below the student name, the subject: with in this case it being Maths. | Functional |
| 11 | Below the subject, the grade: which should Identify as 11. | | Below the subject, the grade: which should Identify as 11. | Functional |
| *Screen shot* | |  | | |
| *Overall comment* | | *Adding students, and all their credentials is working. I plan on removing pictures though, it’s pointless and a HTML5 webapp lacks the functionality to access the devices camera. This is an obvious constraint.* | | |

**Graphing and mark/assessment storage module**

|  |  |  |  |
| --- | --- | --- | --- |
| **Brief description of func** | | **Explanation** | |
| Adds mark/assessment number to database, and graphs them. | | A tutor’s student, has just finished their 3rd assessment, and the tutor wants to know enter the students 3 assessment mark and not only receive a visual demonstration of how the student’s grades have changed but also, be secure in the knowledge that that information is stored. | |
| Pre-condition: | | That student must exist in the database, in order for marks and assessment number to be added to them. | |
| Page descriptor: | | “Student Dashboard | |
| Screen shot | |  | |
| Input data | Expected output | Actual output | Status |
| Mark | Mark added to the database, and added to the chart. | Mark added to the database, and added to the chart. | Functional |
| Assessment number | Assessment number added to the database and added to the chart. On the horizontal axis | Assessment number added to the database and added to the chart. On the horizontal axis. | Functional |

**Storage and display of day/time module**

|  |  |  |  |
| --- | --- | --- | --- |
| **Brief description of function** | | **Explanation** | |
| Adds the day and time of the student’s lesson to the database, and displays it on the page. | | A user wants to change the time of a student, or perhaps the student has just been added and doesn’t yet have a time. They simply select the day from a drop down and enter the time in the field, click save and it stores in the database and displays in the student data section of the dashboard. | |
| Pre-condition: | | That student must exist in the database, in order for a day and time to be added to them. | |
| Page descriptor: | | “Student Dashboard | |
| Screen shot | |  | |
| **Input data** | **Expected output** | **Actual output** | **Status** |
| Day | Day selected now resides in database, and is displayed on the user data column. | Day selected now resides in database, and is displayed on the user data column. | Functional |
| Time | Time selected now resides in database, and is displayed on the user data column. | Mark added to the database, and added to the chart. | Functional |

## Module/Fuction list

List of all program module/functions, with details of functions and features STATUS – what are completed.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Module** | **Date last update** | **Explanation of module** | **Testing completed** | **STATUS: Level of completion, problems etc** |
| Add student | 24/05 | Adds a student to the database, and appends them to a list. | yes | 99%.9999 For some reason, when a student is added they are not appended as a regular jquery list item. They are appended as a large div free listen item. However, when the page is refreshed the newly appended item is now normal. I will try and fix this before due date. |
| Add Mark | 24/05 | Adds the mark data, into an array inside the student object. The mark is also graphed on a HTML5 canvas element. I am particularly proud of this functionality. | Yes. | Fully Functional, and I think it really brings the app to life. Super professional, and actually really handy. |
| Add Day/Time | 18/05 | So, not all of the student data is added to the storage when the student is first entered. In particular the marks, and day/time are entered from the student’s dashboard. This function adds the day/time of the student’s lesson. | Yes | Completed, was not difficult. |
| Delete Student | 23/05 | Obviously students are going to come and go, this functions allows the apps user to delete a student from the list of students and pop them from the database. | Yes | Completed, with yes/no confirmation as well. |
| Edit student data | 22/05 | When you click edit, the editable version of all the students data appears. It can be edited as desired. | Yes | Completed. |
| Undo added mark | 24/05 | When you click undo, the most recently added mark in the array is deleted and the Chart is redrawn with the new array of marks. | Yes | Completed |

# TASK 4 – USER DOCUMENTATION

## License agreement and copyright details

The user of this software, has the right to use, distribute and modify it to their likings. This software is free, and should not be sold by anyone to anyone. If any issues are found with the software, it is your responsibility not to exploit it on other user’s machines without their consent. A breach of any of these terms, may be grounds for the revocation of your software license.

## Copyright license

* Freeware
* OpenSource

## Contact details

|  |  |
| --- | --- |
| Address | 66 Elgin St Gordon NSW 2072 Australia |
| Telephone | +61 447 175 315 |
| Fax | +61 987 654 321 |
| Email | Jacobgreenberg5[@gmail.com.au](mailto:Email@email.com.au) |
| WWW | www.tutraapp.com |

## Installation guide

***System Requirements***

* A modern HTML5 enabled browser.
* 10mb of free storage.

***Locally***

1. *Go to the location you want your software installed.*
2. *Place rar file at location.*
3. *Right click, and select extract here.*
4. *Wait for extraction to complete*
5. *Double click index.html and it will open in your browser*
6. *Or open your browser, and enter the location of index.html, with the prefix of file://.*

***Remotely***

1. *Login to your webserver*
2. *Upload the rar file to the root directory*
3. *Unrar it into the root directory*
4. *Wait for extraction to complete*
5. *On your browser visit your webservers address, it should by default direct you to index.html.*

# TASK 6 – REFLECTION and PROGRAM STATUS REPORT

## Status report

*Program Name: Tutra  
Summary of solution – version 1 as of date 24/05/2015*

Created by: Jacob Greenberg

Degree of success at meeting client/end user needs/problem  
I think I produced very successful results, the client needed a paperless way of tracking the students they teach. I provided an app which does, just that.

Current status of the program

The software is for the most part complete, the UI needs to be optimized for user friendliness. But the functionality is there, it stores user data and displays it in a beautiful chart format.

Features and functions not fully implemented from the stage 2 design

A time table on the home page, which shows which student’s lessons are upcoming.

Future developments of the program including functions and features

I would like to see my application have the ability to export the database, in a format which can be later uploaded and read by the application. Because browser storage has a tendency to be volatile, not to mention it’s common practice erasing it every couple weeks.

Changes and alterations that occurred to your software design from stage 2

Originally, my application was going to have like a timetable on the main page, which not only listed all the students but sorted them by time. This was probably doable, but I spent a significant amount of time working on my graphing. That still is I think the best part of the application. There was also originally going to be a login page, but It just felt unnecessary and was me just vying for more modules. To make the documentation easier.