Final Report

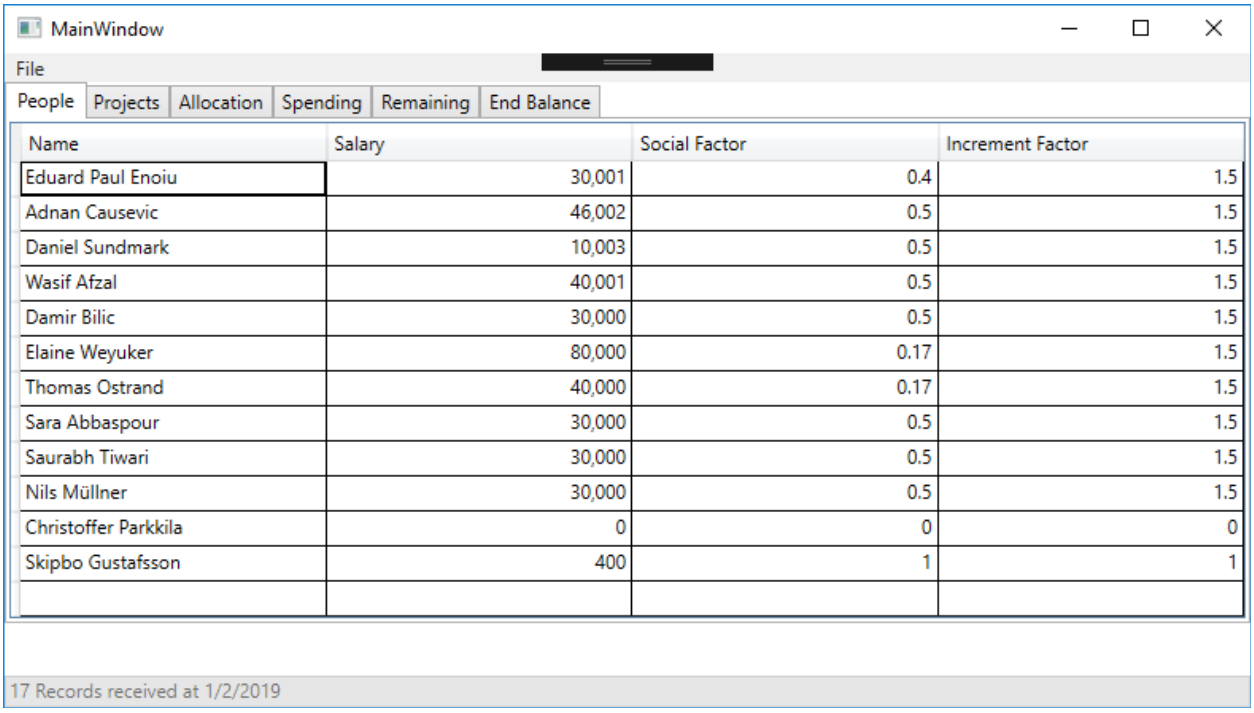
Mälardalen University
Academy of Innovation, Design and Engineering
Project name: Graphical Project Portfolio Management Web Application
Project group: 3
Course: DVA313 – Software Engineering 2
Publication date: 2019-01-17

Table of Content

1. Introduction

At the beginning of this course, the group were introduced to Daniel Sundmark, which is the client for this project. The client is mainly working with handling several projects at Mälardalens University. Included in this work is allocating staff and keeping track on expenses of these projects. To administrate this he has been working with a desktop application developed to match his needs. However, the needs has changed and they would prefer a more user-friendly design. In addition to the user-friendliness, they wanted, instead of a desktop application a web application.

The original desktop application (see Figure 1 and 2) is not anywhere near neither user-friendly nor up to date with what an application would look like today. It consists of six different tabs; people, projects, allocation, spending, remaining and end balance. The people tab would list all of the people on all projects. The project tab lists all projects that Daniel has entered. Allocation tab shows how much each person work on each project. Spending is how much each project costs (considering staffing and other expenses) and remaning is what is left after spending. End balance shows what is left of the money when project is done.



Name	Salary	Social Factor	Increment Factor
Eduard Paul Enoiu	30,001	0.4	1.5
Adnan Causevic	46,002	0.5	1.5
Daniel Sundmark	10,003	0.5	1.5
Wasif Afzal	40,001	0.5	1.5
Damir Bilic	30,000	0.5	1.5
Elaine Weyuker	80,000	0.17	1.5
Thomas Ostrand	40,000	0.17	1.5
Sara Abbaspour	30,000	0.5	1.5
Saurabh Tiwari	30,000	0.5	1.5
Nils Müllner	30,000	0.5	1.5
Christoffer Parkkila	0	0	0
Skipbo Gustafsson	400	1	1

Figure 1 - People tab view

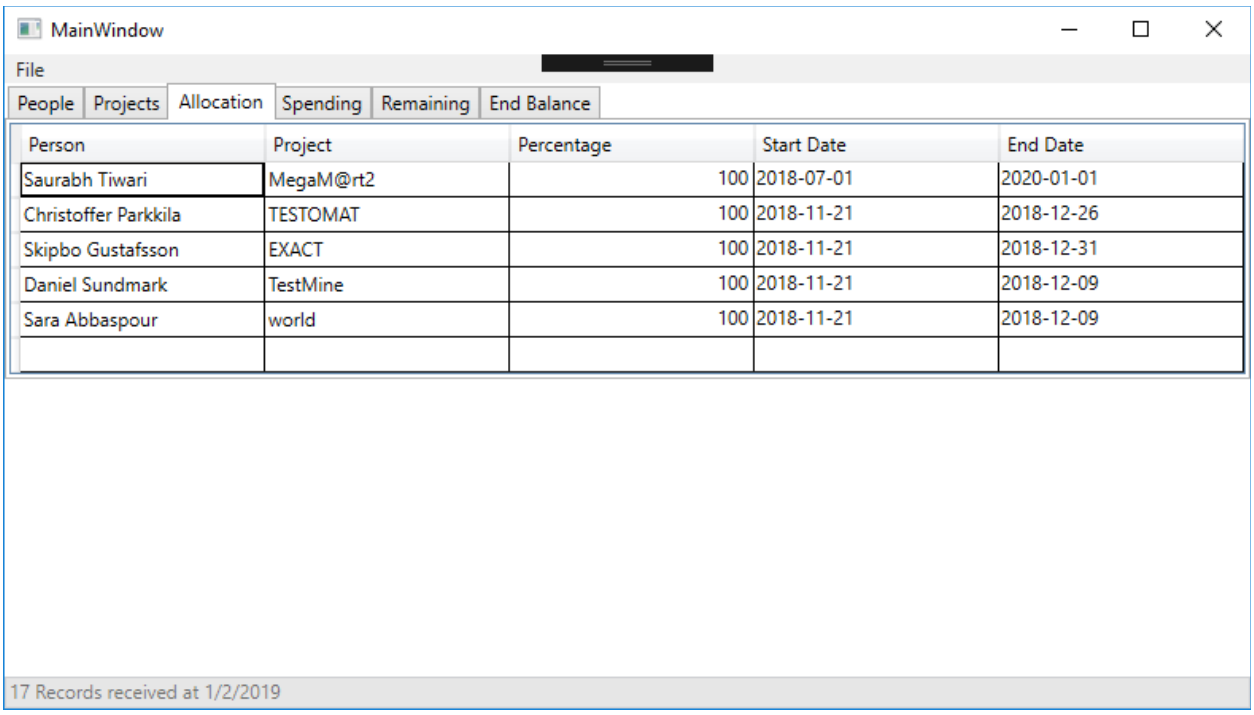


Figure 2 - Allocation tab view

1.1 Product to Deliver

To ease the clients work, the group were to develop a user-friendly web application where the client could in a modern, graphical user interface (GUI) manage their staffing. The main focus were on the allocation tab where the group were to implement a timeline representing a horizontal calendar. On this timeline the client wanted to be able to add new allocations. This allocation should be added just by clicking on the timeline and it would appear straight away. The allocation should be draggable horizontally to set the duration of the project. The percentage was at the beginning supposed to be dragged vertically to change. However, due to clients request, they wanted to change this because they realised that it might look bad on the timeline and therefore, percentage will be entered manually on the allocation. Furthermore, the client wanted to be able to list all staffing next to the allocation making it easy to swap between people.

During the development phase, the client was involved in almost every decision which has been taken. For example, new features to the allocation tab was added such as; when person is allocated more than 100%, the tab should show red or when adding new items such as persons, projects or allocations, the data should not be sent to database unless the save button is pressed.

1.2 Requirements

Following diagram will show the requirements we had at the start of the project, sorted by importance. They will also give a further understanding of what was expected of the application and will complement what was mentioned in section 1.1.

ID	Description
1	One must be able to select a person from a list of persons to display that persons allocation view
4	One must be able to double click on a persons allocation view to initially allocate some amount of time
5	One must be able to click and drag the allocation horizontally to allocate time
6	One must be able to click and drag the allocation vertically to allocate the employment rate
7	One must be able to break an existing allocation to specify different employment rates for different time periods
8	The allocation view must contain all the projects that the user is currently working on
10	The allocation view must contain a total timeline that displays the summation of all the separate allocations
3	The allocation view must contain some calendar or timeline in the background for the individual projects
9	The allocation view should only contain projects with active allocations
11	The allocations should be snapped automatically to the end of each month
12	The allocation view should be zoomable to facilitate the allocations and overview for a specific project
13	The system must ensure that a system user cannot allocate time that exceeds the projects end date
14	The system must ensure that a system user cannot allocate an employment rate that exceed full time
2	One must be able to search for a user in the persons list
15	The system should be able to generate a report over persons allocation that a system user can save in a certain file format
16	A view over a specific project that displays all the persons and their allocations would be a nice future to have
17	One must be able to add a person in the persons tab
18	One must be able to edit a person in the persons tab
19	One must be able to remove a person in the persons tab
20	One must be able to add a project in the projects tab
21	One must be able to edit a project in the projects tab
22	One must be able to remove a project in the projects tab

Table 1 - System requirements

1.3 ReactJS Framework

After researching online, some members found that ReactJS is one of the most attractive frameworks on the market and it would suit the needs of the client very well. ReactJS is a JavaScript framework developed by Facebook. It is used to create dynamic, aesthetically appealing user-friendly applications, both for web and smartphones. Using ReactJS resulted in that the front-end side of the project only consist of ReactJS elements. It made it possible to create a visually appealing and coherent application. In addition to this, we added several ReactJS-component such as the navigation bar, the tables and the timeline which all was possible to manipulate into what the client wanted.

2. Final Product

The product we delivered at the end of the project turned out as a react-based web application. Just like the original desktop application, our application contains the same tabs, with additionally one tab called ”Save File” which is letting the user save the projects in different file formats. Furthermore, compared to the original application, the new one is far more visually appealing and has a more modern touch. For example, when looking at the people tab in the original application (*see figure 1, p. 3*), it is hard to see straight away where to add new staff. To add new staff, the user has to enter the information in the empty text-field. To reduce these misunderstandings we added an add-button. We also made it possible to delete people with a button next to the add (*see figure 3*).

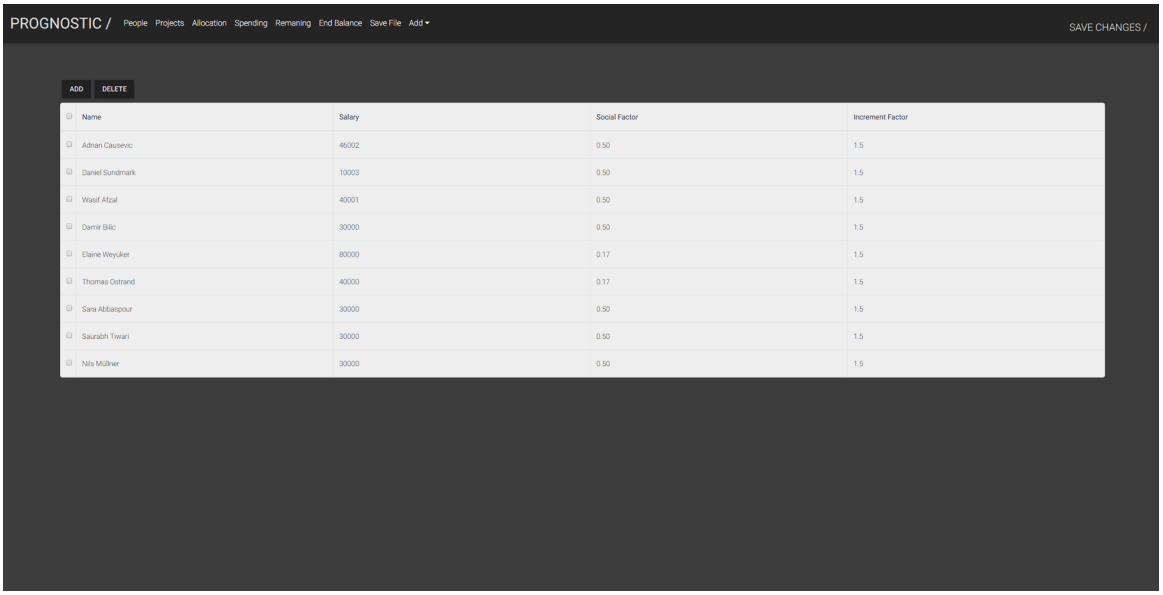


Figure 3 - New people tab

2.1 Final Product Features

The front-end consists of several React components combined. Most of these components are connected to a database which was set up at the beginning of the project. The database is set up as a MariaDB which contains of several tables which we got from the client. The connection is made through an implemented PHP controller which fetches data from the database through several PHP files, making it possible for the React element to access it. There are also calculations (i.e. to calculate end balance) implemented in the back-end.

To navigate around the application, it uses a navigation bar (*see figure 4*). In the navigation bar it is possible to browse between all the different tabs, but it also contains a global save-button at the right hand corner. When pressing this button, all new entered data will be sent straight to the database. When entering new data on any of the tabs, it will only be stored locally until the save button is pressed.

Figure 4 - Navigation bar

2.1.1 People and Project Tab

The first two tabs found in the navigation bar are the people and the project tab. Both tabs contains a table filled with either staffing or projects. Above the tables the user will find two buttons, one for adding a new person or project and the other one for deleting one or several persons/projects. To add a new person/project the add button is pressed. When the button is pressed, a pop-up appears where the user can enter all information required. To delete one or several people/project the user will find boxes on the left side of the table where they could tick them, and then press delete. It is also possible to tick the box in the title row to select everything in the table at the same time. This covers the ID's 17-22 in the list of requirements.

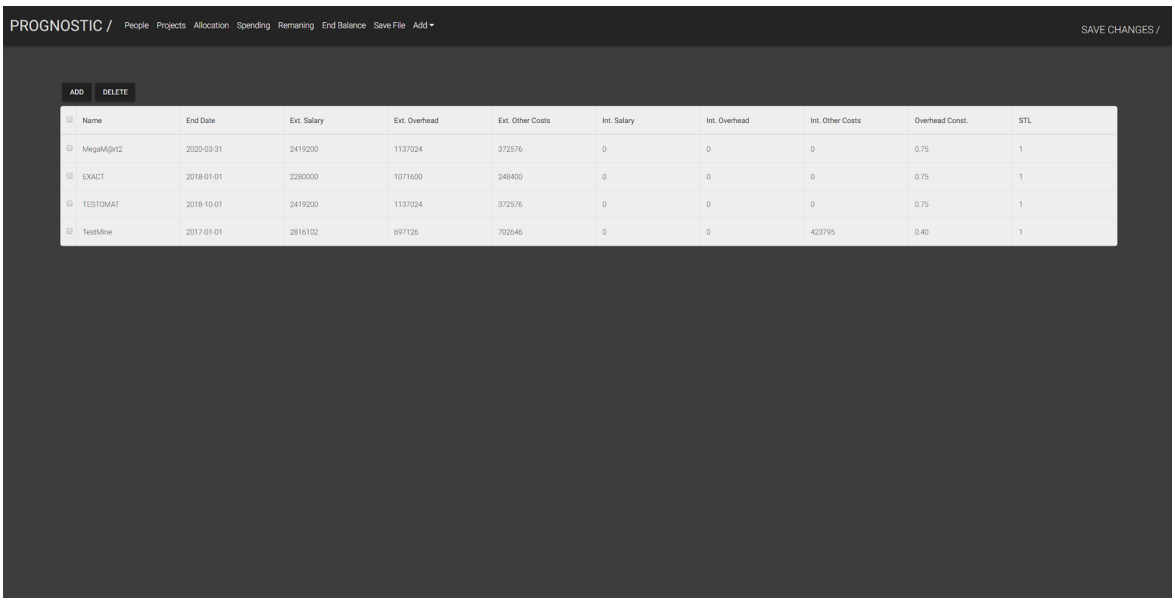


Figure 5 - Project tab

2.1.2 Allocation Tab

The component which is the most complex is the allocation tab (*see figure 6 on next page*). On the left hand side, the user can search for all different people who are to be found in the database. When pressing their name, the allocation timeline will appear. The timeline will list all projects on the left, even the projects whom the person is not allocated on. The user can hide non-active projects with the toggle-button. At the bottom of the timeline, the duration is listed. When user chooses to zoom either in or out, the duration will either be over a longer time, such as displaying years or when zooming in, the timeline will be more detailed into specific weeks.

To declare a new allocation for a person, the user will have to double click on the specific line for the desired project and an allocation will appear. The default duration depends on how much the timeline is zoomed at the moment. To change the duration of the allocation, the user can drag the allocation in both horizontal directions. To remove a certain allocation the user has to press it and in the right corner, a remove button will appear. Pressing this and the allocation will disappear. To edit the percentage a person should be working on project the user has to press the numbers on the allocation and change them manually by enter figures. If one person is allocated on a project which already has passed its due date, the allocation will automatically change its colour to red. In addition to this, the user could easily split an allocation by right-click the allocation.

At the top of the timeline, the total time is displayed and visualised with colours and height; if a person is allocated less than 25%, a darker green will appear, 25% - 50% is a medium intense green colour and a lighter green when the allocation is anywhere between 50 - 100%. If the allocation overrides 100% but still remains under 150% yellow tones appears. If the allocation exceeds 150%, the colour will be a darker shade of red.

Regarding the requirement list, the implemented allocation tab covers following IDs; 1, 4, 5, 6, 7, 8, 10, 3, 9, 12, 13, 14, and 2.

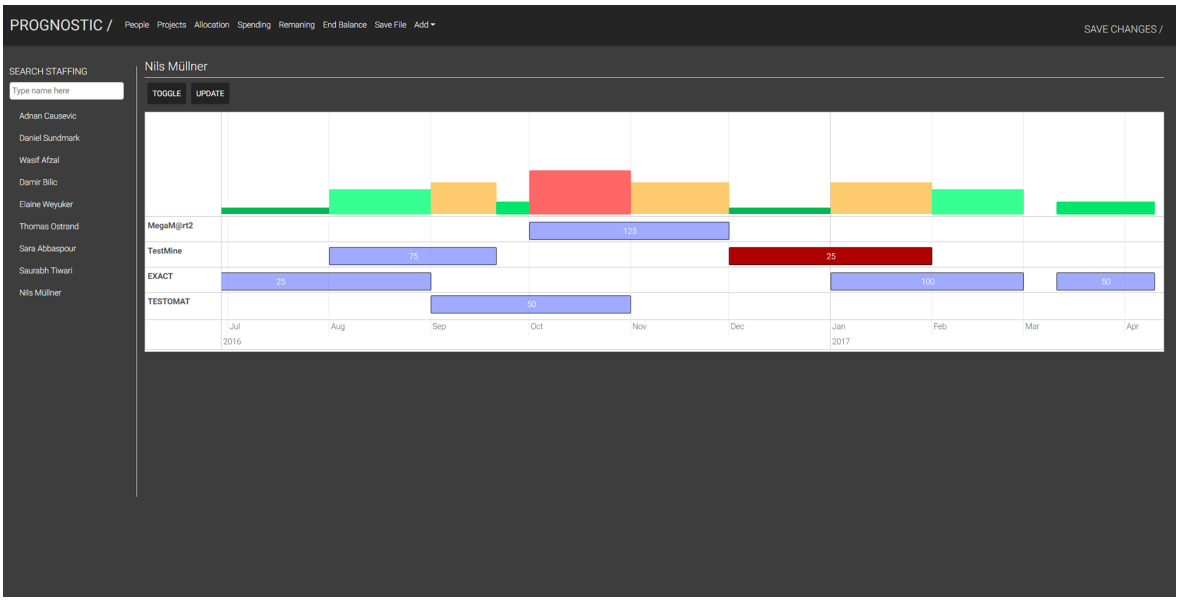
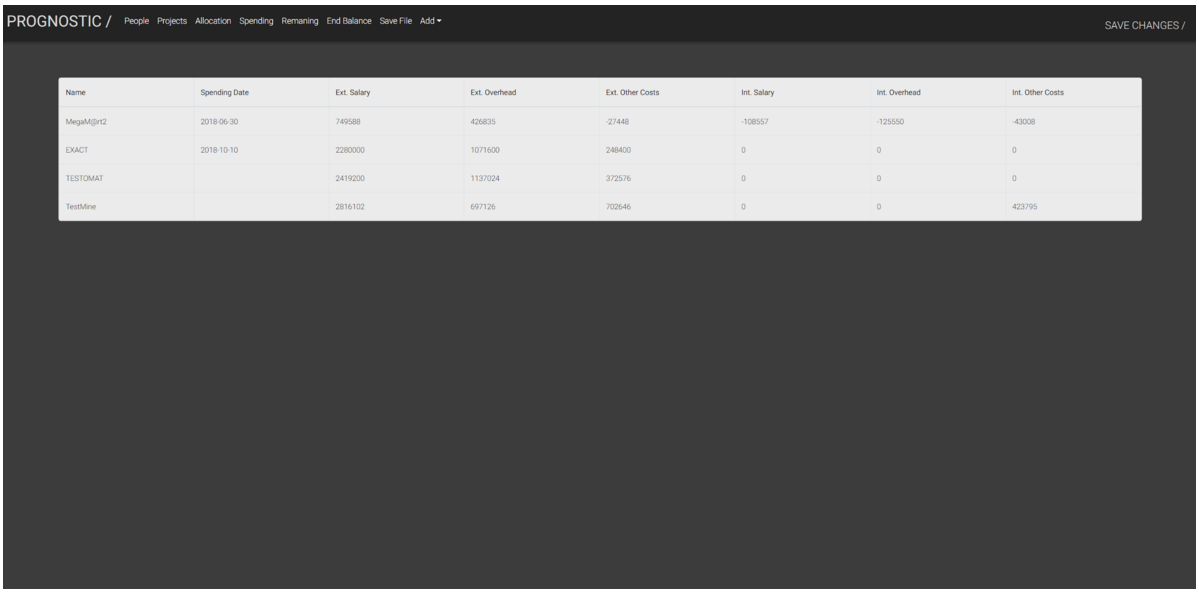


Figure 6 - Allocation tab

2.1.3 Spending & Remaning Tab

These two tabs are very similar to each other. They do remind of the people- and project tab. These

ones however, do not contain any buttons for adding and removing. These pages only consist of tables displaying the specific data fetched from the database for the specific tab, just like the original application. However, it is not plausible to change data in the spending tab manually. It is also possible to sort the projects by different values.

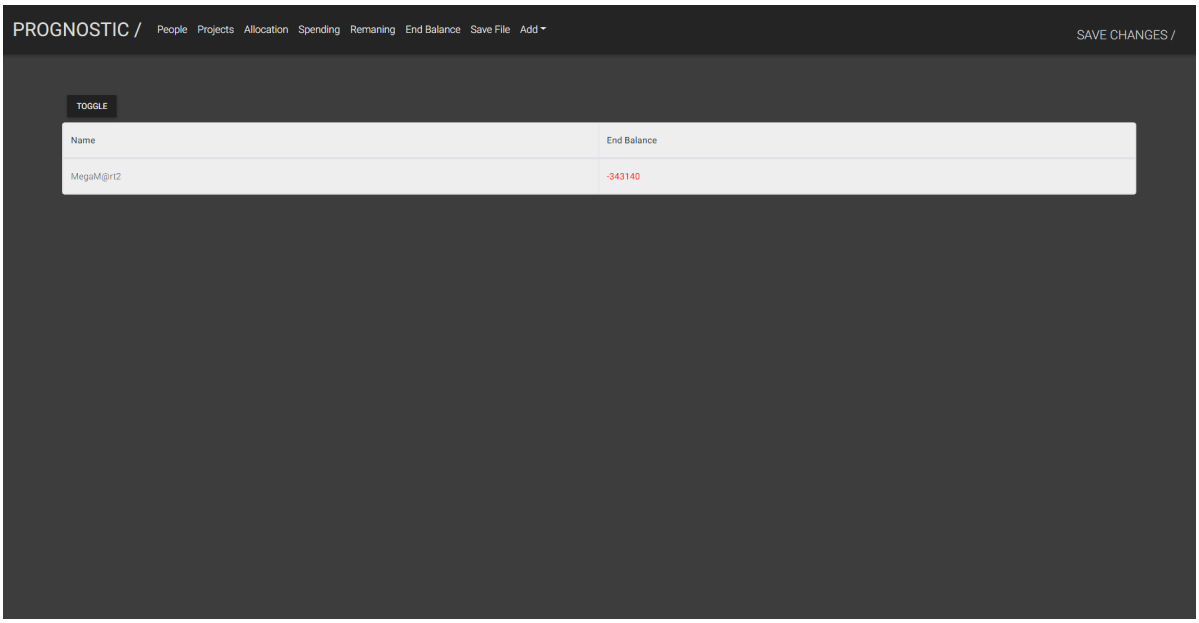


Name	Spending Date	Ext. Salary	Ext. Overhead	Ext. Other Costs	Int. Salary	Int. Overhead	Int. Other Costs
MegaM@rt2	2018-06-30	749588	426835	-27448	-108557	-125550	-43008
EXACT	2018-10-10	2280000	1071600	248400	0	0	0
TESTOMAT		2419200	1137024	372576	0	0	0
TestMine		2816102	697126	702646	0	0	423795

Figure 7 - Spending tab

2.1.4 End Balande Tab

The End Balance tab consists of a table and a toggle button. Before toggle button is pressed, the user will only see what remains after project is finished. If the figures are red, it means that the project spent more money than expected and has a negative value. When the toggle button is pressed, all other data will also appear. Behind the scenes there are calculations which calculate the end balance in the database. This table is also possible to sort the tables by different values.



Name	End Balance
MegaM@rt2	-343140

Figure 8 - End Balance tab toggled (first state).

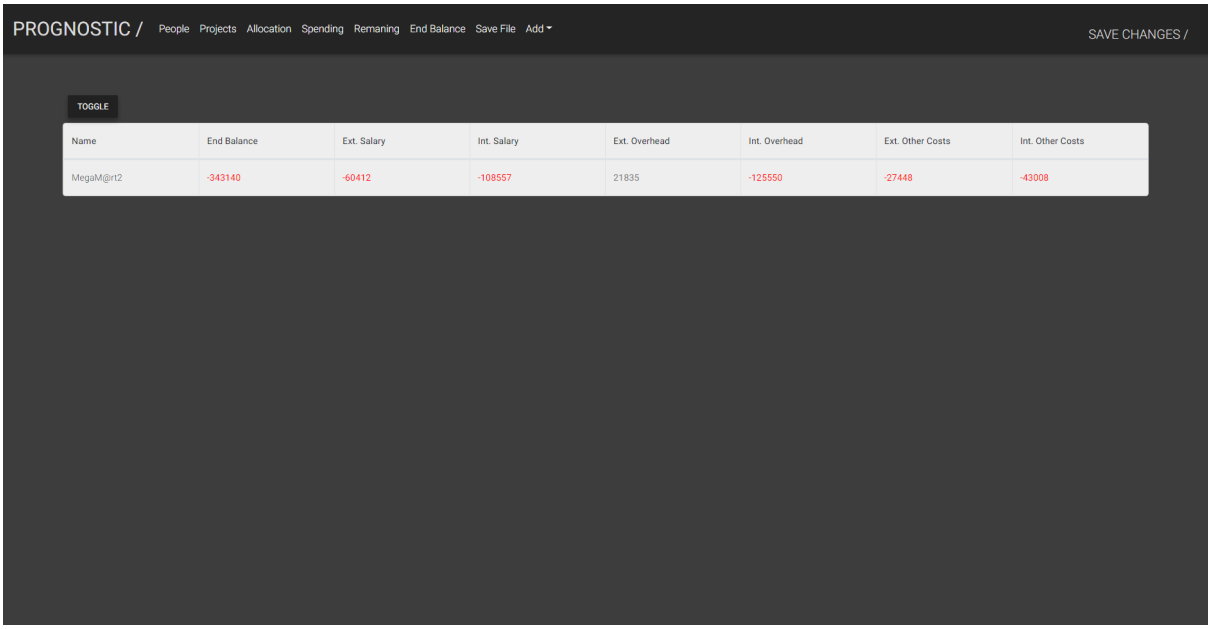


Figure 9 - End Balance tab not toggled.

2.1.5 Save File Tab

This tab is supposed to make it possible for the user to save project information to a PDF. The tab will show a table listing all existing projects. From here the user will be able to select one or several projects and save into chosen file format. This covers ID 15 in list of requirements. However, due to lack of time this is not fully integrated in the GUI, but the functionality is already implemented.

2.2 Not Implemented Features

Starting from the list of requirements, the only thing not implemented is ID number 16. It would be a nice feature having a view which displays all the allocations of a certain project. However, due to lack of time there was no chance of implementing this. Another thing is one feature which was not intendedly in the original plan was to have. In the navigation bar, it was supposedly to be a button with a drop-down menu containing quick-adds for adding either people or project without having to enter the tabs. This idea was also dropped due to lack of time.

2.3 Big Changes From the Client

Since the day the project started, the client has been very clear in their wishes. On top of that, the project is based on developing a better user interface for an already existing application. This resulting in that there never were any major changes from the client along the way. However, one thing worth mentioning - after the start of the project the client mentioned that they wanted to be able to drag the allocation vertically to change the allocation of person. In addition to this, they wanted some colour or any other visuality to show when allocation is over-allocated.

2.4 Acceptance test

To test our system, we firstly performed a black-box test with the client. We gave no other instructions rather than explaining what the system does and what the functionalities are (i.e. letting the client know what certain tabs does etc.). The client moved around the website as they would preferably use it. There were no specific issues identified during this test and the client seemed very pleased with system, apart from small changes in the GUI. However, this did not affect anything on how to actually use the system. In the Appendix at the end of report, all test cases are listed.

Since we needed five more days to work on the project, we also performed a proper acceptance test following the acceptance test cases found in the appendix. This test was done by two of our own team members. In order to inform and update our client on the test cases, they received a summary of the test cases similar to this section of the report. The acceptance test file is an excel document consisting of six tabs; staff table(st), search function(s), nav bar(n), tables(t), project(p) and allocation (a). Worth mentioning is that minor changes to prevent some of the fails and bugs, changes has been made after the acceptance test. However, due to lack of time another acceptance test was not possible.

2.4.1 Staff Table (ST)

This section validates the table values and checking if invalid inputs are accepted, verifying if the user is able to modify present data and verifying wether or not the user is able to save the new changes. Most of the cases was passed, however, it turned out it was possible to enter invalid data when adding new person.

2.4.2 Search Function (S)

To test how the search function in the allocation view works this section covers test such as verifying if the user is able to enter data in search field, see if the search result is correct - i.e showing the person whom is searched for if they exist or listing none if the person does not exist. All of the test-cases were passed.

2.4.3 Navigation Bar (N)

To cover how the navigation bar functionality works there are test which verifies if all the items are displayed in the navigation bar, if the user can switch between different items in the bar and also verifying if the user is directed to the expected page when specific item is pressed. This also passed all of the acceptance tests.

2.4.4 Tables (T)

This sheets includes the test cases related to all of the tables besides the project table. This section verifies if the user can view the table correctly but also verifying that the data is displayed in the tables as accurate. Every test case passed.

2.4.5 Project Table (P)

This section validates the specific values for the project table. In other words, validating the table values, checking if invalid inputs are accepted, verifying that the user is able to modify the already existing data and if it is possible to save new changes. This had similar output as the staff table; it was possible to enter invalid input. However, modifying and adding section worked fine.

2.4.6 Allocation View (A)

The allocation view is the most complex part of the application. To cover everything in testing following sections are tested; verifying if the user is able to view all the contents in the allocation view, check if the zoom in and out is working properly and changing to the right dates and values, verifying that entered data is correct and displayed accurate in allocation, verifying wether or not the user is able to delete allocation and verifying that if project is past end date, it should give warning. All of the test ware passed in this case.

3. Working as a Group

At the beginning of the project, we as a group found it hard to divide the work amongst eight people. However, since the project needed both people working on the back-end and the front-end, we split the group equally, at least for the beginning of the project. This resulted in Christoffer, Erika, Filip and Sai started implement the front-end whereas Matko, Mohammed, Osamah and Zaid sorted out the back-end with the database and such. In addition to this, most of the members got extra responsibility. However, everyone did not take on extra responsibilities. In sum, the original plan looked as following:

Christoffer Parkkila: Developer with responsibility for front-end, GitHub and also the time reporting.

Erika Weilander: Project manager, client contact and developer with responsibility for front-end development, contact with steering-group and client, taking care of Trello-board.

Filip Andersson: Developer with responsibility for front-end and all PowerPoints.

Matko Butkovic: Developer with focus on back-end (left at end of course).

Mohammed Abusamaan: Developer with responsibility for back-end and testing.

Sai Vijay Vemasani: Developer with responsibility for front-end and taking notes.

Osamah Al-Braichi: Developer with responsibility for back-end and documentation creation.

Zaid Aber Jaser: Developer with responsibility for back-end and documentation creation.

3.1 Changes of Organisation and Routines

Since we set roles and responsibilities before we even started with the development of the application, the responsibilities came to change during the process. One of the major changes were a few weeks into the development when the back-end reached its endpoint. Then we moved three people from the back-end development to the front-end development. This was also because the allocation time-line did take more time than planned and it needed more people working on it. However, this was also needed because at a point we had to work on the connection with front-end and back-end and the people most suitable for this would be the ones working on both teams. In addition to this, Sai and Matko swapped responsibilities and Matko became responsible for taking notes at meetings. At end of project, due to lack of time Sai took over the validation and verification responsibility from Mohammed.

3.2 Total Project Effort

At the beginning of the project we tried to decide how to divide the effort, but at the end we found it very hard. Instead we decided to decide what to do every week and therefore, these numbers are based on the hours and comments we have in our time report. The figures are rounded up to closest fifth percentage. The category miscellaneous (misc.) includes meetings, lectures, administrative etc.

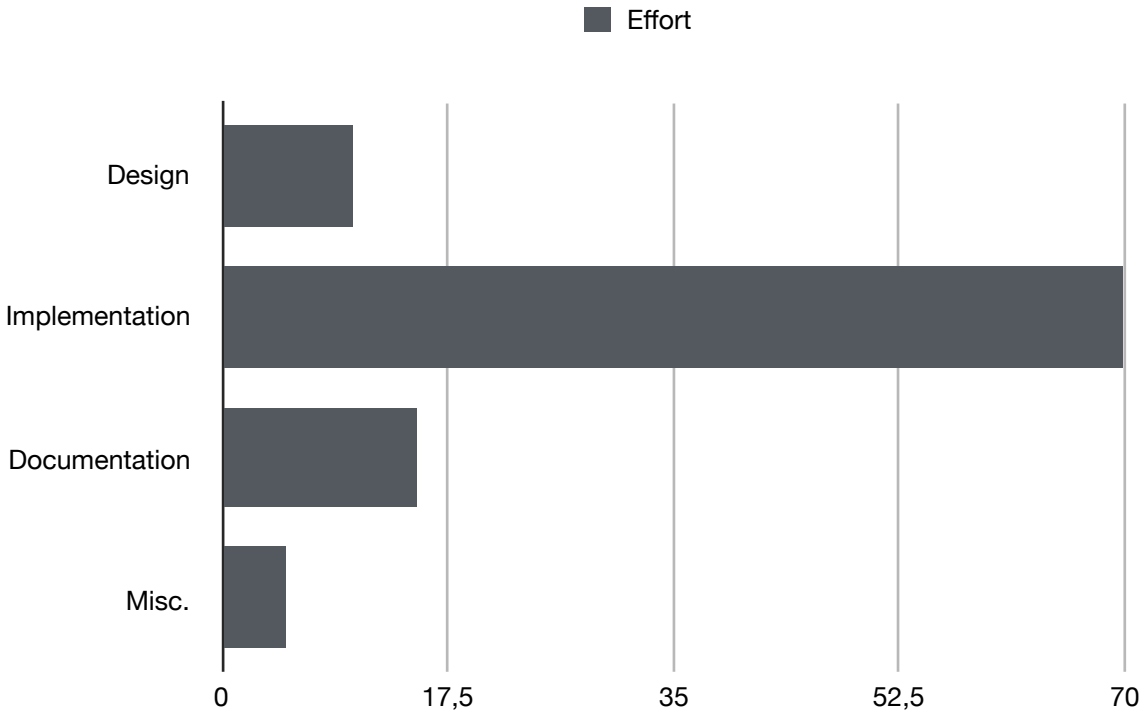


Diagram 1 - Effort in percentage

3.3 Individual Effort

To give a clear and fair view of what everyone has been doing, following text is based on the responsibilities people has been taking on Trello, but also what group members has reported in our excel file where the time reporting is done. There are few activities we have done together such as writing the project plan and the design document due to bad planning. Together, depending on whom doing the presentation, people have prepared presentation. Everyone was responsible for reviewing this document, however everyone did not.

Christoffer Parkkila: At the beginning of project, Christoffer set up the time-report file, the project plan template and also begun to read up on GitHub. He created both a template from which other group members could access and understand GitHub. He also held an informative GitHub lecture for the group. Along side the other people working with front-end he spent time researching what frameworks to get the best solution for the application. When the implementation phase came along, Christoffers main responsibility has been working with the allocation time-line. His time was mostly spent on implementing it and manipulating it into what the client would need, such as: visualisations, total time, split allocation when right clicking etc. Besides his own work he has been putting a lot of time into helping others. For example, getting people started with Git, starting the project, explaining how the PHP should work etc. Since we decided to work on the project 5 days extra Christoffer managed to; fix several bugs, changed toggle-button, wrote a guide on how to deploy project for the client, wrote an pdf regarding the problems between front- and back-end, and test the application. Overall, Christoffer has been responsible for merging everyones code and making sure that the development branch is up-to-date.

Erika Weilander: Since Erika is the project manager, she has spent a smaller amount of time on both emailing with both client and steering-group. Also responsible for Trello and been making sure it is up to date and that cards are moving forward during the process. Since Erika is also a part of the front-end group, she as well has been looking into different frameworks which could help with the implementation. Her main responsibility within the development has been the navigation bar which makes it possible to switch between all components. In addition to this, she has been responsible for the UX and the UI-structure of the application. By comparing similar ideas on the web, she structured a modern, easy and understandable design. She has also taken on a lot of responsibilities when it comes to the written documents; writing, re-structuring, editing and correcting. For example, she has written most of this report, except section 4 & 5. At the end of project she has been looking into how to make the navigation bar work during deployment. This she continued with after the end of project. She also wrote a PDF on how the test-cases went to provide the client with. In addition to this, she updated this report with all new information.

Filip Andersson: Before every meeting and every presentation, Filip has provided the group with well-structured PowerPoint-slides. Apart from the PowerPoint responsibility Filip has been working with all the different tables to be found in the application. He also made it possible to add new people and project in the specific tabs by letting the user entering information in a pop-up window. He also made it possible for the user to remove projects and people. Since Filip is also part of the front-end team, he also researched for different frameworks at the beginning of the project. At end of project, Filip wrote a structured guide on how to install the project for our client but also

added the function to toggle the end-table values (i.e hide all tabs besides the final end-balance) and made it possible to the sort tables by values. He also wrote section 4 & 5 for this report. After the final presentation Filip worked with researching the best solution for input valids for the tables and implemented the solutions he found most useful.

Matko Butkovic: Matko is the most recent member of the project and came in to the project a week later than everybody else. Since then, he has been responsible for taking notes at meeting but also a part of the back-end team for two weeks before moved to front-end team. At current we have no information of what Matko has been doing apart from catching up on missed meetings and reviewing and helping out with the first report. Matko left at end of project without any notice.

Mohammed Abusamaan: Mohammed has been our main back-end developer. With experience from databases earlier, he, together with Zaid and Osamah set up a working database. Furthermore, Mohammed has made it possible to send and receive data from the database. When done with database he has spent a lot of time working with connections with AJAX, JSON and PHP and been trying to create connections between front-end and back-end. At end of project he also became more responsible for the PHP connections together with Osamah. In addition to this, Mohammed has been making several diagrams for our papers and integrated the client calculations in the database.

Sai Vijay Vemasani: Since Matko took over the responsibility for taking notes, Sai has been mainly been a front-end developer. He has been responsible for working with implementing the fundamentals of the search-bar on the allocation-tab. Apart from that Sai has been looking into different frameworks like the rest of the front-end group. At the end of the project, Sai and Osamah worked together trying to connect the front-end with back-end via PHP before Mohammed took over for Sai. After that Sai worked with writing the test cases for the acceptance test, he also performed test-cases testing the program according the document he wrote.

Osamah Al-Braichi: Osamah has been responsible together with Zaid to create templates for the three documentation deliverables, making sure they were all ready to just fill with text. Apart from that, Osamah has been working with Mohammed and Zaid with the back-end but also working a lot with connecting the back-end to the front-end with several javaScript classes and different PHP-files. After the final presentation Osamah has been helping Zaid trying to merge the Save to File function to the rest of the project. He has also been working with testing the application according to the test cases Sai provided in the Appendix.

Zaid Aber Jaser: Together with Osamah, Zaid has been responsible to create templates for the three documentation deliverables. Alongside this Zaid has been working with Osamah and Mohammed with back-end related work, such working with database etc. At the middle of project, Zaid switched to front-end and he was the one who made it possible to save the data from tables into different file-formats.

3.4 Project Members Working Hours

To keep track on how much each person has been working, we created an excel-file which each person were responsible to fill out how much time they spent every week on the project including

comments on what they have been doing. Therefore, these numbers are based on the specific time which has been reported individually (*see figure 10 on next page*). At the bottom of the figure it says how much the group has worked together. To get one persons actual time spent on project this need to be added to each individual.

NAME	W1	W2	W3	W4	W5	W6	W7	W8	W9	W10	TOTAL
Erika Weilander	4	10	14	17.5	10.5	20	0	18.5	32	10.5	137
Christoffer Parkkila	4	12	14	18.5	14	13	5	17	15	20.5	133
Zaid Abed Jaser	5	5	8	6	9	15	2	20	14.5	2	86.5
Filip Andersson	3	3	5.5	18.5	6.5	4	0	9	23	9	81.5
Matko Butkovic	10	6	13	10	2	0	0	0	0	0	41
Osamah Al-Braichi	4	7	9	10	11	10	10.5	13.5	10.5	4.5	90
Mohammed Abusamaan	8	4	14	7.5	17	10	1.5	19.5	29	0	110.5
Sai VijAy	3	4	14	11	8	11	11	15.5	15	3	95.5
Group	4.5	7.5	6	2	7	3	0	2	1.5	5	38.5

Figure 10 - Time Reporting

4. Lessons From This Project

The project has provided us with several positive experiences and insights that is useful for the future, either in further projects or work. The project provided us the possibility to practice working in groups. Learning to work cooperatively with different people as a team is a valuable trait that is useful in many professions, let alone computer science and software engineering. When working as a team, communication is key, and so is planning. Both of these are skills that requires practice and experience, which this project has provided for us and given us a chance to expand upon. But working in a group has its downsides as well, especially such a large group, as it is harder to manage, keep everyone up do date and plan meetings. The motivation and goals for each member is also varying, and there is a clear difference between those who are dedicated to the project and those who are uninterested. However, despite being downsides, they are still useful experiences to learn.

The project let us put the things we have previously learnt to test, both from the prerequisite course *DVA332 Software Engineering 1: Basic Course*, but also other courses provided by the university that came in handy and were put to use. We were given the opportunity to make use of all the theoretical elements and ways of working learned in the basic course and divide the work and responsibilities among the group members to see how the perform in practice. We also got the experience to work in sprints - to divide work in smaller parts and have more common deadlines, checkups and syncs. This was a very effective way of working, as it was easy to keep track of who is carrying their weight during each week. It also causes less backlash if someone has not finished their task compared to several weeks of working, and help can be provided earlier in the process.

The project has also provided us with the experience of working with a real client, something that was new to most of us. It was motivating to work with a real client, because we were working for someone expecting a good result that we did not want to let down, and we wanted to prove what the group was capable of, and provide a product exceeding the expectations. And once again communication is key here, as regular meetings or other sorts of checkups are crucial to

make sure the work is on the right track. We also learned that despite being provided with a lot of feedback from the client, it is important to ask for it as well. By analysing potential options and providing them to the client well in advance, it gives the client more time to think them through.

Besides everything mentioned above, we have also got new knowledge of useful tools such as React, that can be of use for further works in the future. Most of us also did not have much experience with code repositories like GitHub, which this project also helped us exceed in and get better practice of. Both GitHub and Trello were great for communication and cooperation and were helpful to make sure everyone were on track.

As with most projects and assignments, it is always easy afterwards to have better insight of how the work should have been distributed differently. With the experience that we have now, we know which parts took the most time and would have benefited from more attention or being put to work earlier. We should also have been stricter on the internal deadlines of the sprints, but it should also have been up to each member to really put in the effort to solve their assigned task and ask for help before the end of the sprint if needed. A third thing we should have done differently would have been to finish more work before Christmas, as the holidays steals a lot of attention from the project and people become less active.

5. Advices for Upcoming Students

One advice we could give to other groups is to try dividing the work so that each member gets a part that they enjoy, but also is at least somewhat experienced in. The reason for this is because it's more motivating to work with something you enjoy, and while it can be both useful and fun to learn something new, there are deadlines that needs to be met and it's likely to be more efficient to distribute the existing experience where it is needed the most. Another advice is to keep up with the communication and always be honest. Do not hide away if you are not able to deliver your assigned task. Ask for help - the sooner the better.

Appendix

Acceptance test-cases

Test Scenario ID	Test Scenario Description	Test Case ID	Test Case Description	Test Steps	Preconditions	Test Data	Post Conditions	Expected Result	Actual Result	Status	Executed By	Executed Date	Comments
ST_001	Add new person to staff table	STTC_001	Enter valid entries	1. press add button 2. enter valid values in the fields 3. press save button	Test Data Valid URL	5 samples of Random valid values	message "Added Successfully "	popup "Added successfully"					
		STTC_002	Enter invalid enteries	1. press add button 2. enter invalid values for each field 3. press save button	Test Data Valid URL	5 samples of Random invalid values	invalid inputs message	popup "invalid inputs"					
		STTC_003	Enter invalid enteries	1. press add button 2. enter invalid values for one or two fields 3. press save button	Test Data Valid URL	5 samples of invalid values	invalid inputs message	popup "invalid inputs"					
		STTC_004	No Entries	1. press add button 2. No values Entered 3. press save button	Test Data Valid URL	No entries	No inputs message	popup "No inputs"					
		STTC_005	Enter special charatcers	1. press add button 2. enter special characters for each field 3. press save button	Test Data Valid URL	5 samples of inout including special characters	invalid inputs message	popup "invalid inputs"					
		STTC_006	Verifing that the Name field in the table doesnot accept invalid values	1.Click on add Staff button 2.ENter valid inputs in the fields except for the Name field 3.Click on the Submit button	Test Data Valid URL	Three valid and one invalid entries	invalid inputs message	popup "invalid inputs"					
		STTC_007	Verifing that the Salary field in the table doesnot accept invalid values	1.Click on add Staff button 2.ENter valid inputs in the fields except for the Name field 3.Click on the Submit button	Test Data Valid URL	Three valid and one invalid entries	invalid inputs message	popup "invalid inputs"					
		STTC_008	Verifing that the Social Factor field in the table doesnot accept invalid values	1.Click on add Staff button 2.Enter valid inputs in the fields except for the Name field 3.Click on the Submit button	Test Data Valid URL	Three valid and one invalid entries	invalid inputs message	popup "invalid inputs"					
		STTC_009	Verifing that the Incrementalfield in the table doesnot accept invalid values	1.Click on add Staff button 2.Enter valid inputs in the fields except for the Name field 3.Click on the Submit button	Test Data Valid URL	Three valid and one invalid entries	invalid inputs message	popup "invalid inputs"					
ST_002	Modifing the data of a particular person in the Staff Table	STTC_010	Selecting a person in the Staff Table	1.Click on the Person row which needs to be selected	Test Data Valid URL			Should be able to select the person row					
		STTC_011	Modifing the present data in the Staff Table with valid data	1.Click on the field to modify 2.Delete the existing data 3.Enter valid data 4.Save the changes	Test Data Valid URL	valid data		Should be able to modify the data present in that field					
		STTC_012	Modifing the present data in the Staff Table with invalid data	1.Click on the field to modify 2.Delete the existing data 3.Enter invalid data 4.Save the changes	Test Data Valid URL	invalid data		popup "invalid inputs"					
		STTC_013	Modifing the present data in the Staff Table with Null data	1.Click on the field to modify 2.Delete the existing data 3.Save the changes	Valid Url	No entries		popup "invalid inputs"					
ST_003	Saving the data	STTC_014	Verifing that the data is saved in the database	1. press add button 2. enter valid values in the fields 3. press save button 4. close and reopen the application. 5.Click on the People table from the Nav bar	Test Data Valid URL	valid data		The data should be saved correctly in the database					
		STTC_015	Verifing that the Modified data is saved in the database	1. press add button 2. Modify the the data in the table with valid data 3. press save button 4. close and reopen the application. 5.Click on the People table from the Nav bar	Test Data Valid URL	valid data		The data should be saved correctly in the database					

Test Scenario ID	Test Scenario Description	Test Case ID	Test Case Description	Test Steps	Preconditions	Test Data	Post Conditions	Expected Result	Actual Result	Status	Executed By	Executed Date	Comments
S_001	Search Staff	STC_001	To check if search check box takes input	1.Click on the search box 2.Input a string into the box	Test data	Any string value		should be able to enter the string					
		STC_002	Enter Spaces in Search Valid	1. Enter the Spaces in the search field	Test data	Spaces		NO search result should be displayed					
		STC_003	Enter Special Characters in Search field	1.Click on the search box 2.Input Special characters into the search field	Test data	Special Characters		NO search result should be displayed					
		STC_004	Enter Numbers in the Search field	1.Click on the search Field 2. Input Numerical values into the Search field	Test data	Numerical Values		NO search result should be displayed					
		STC_005	Check if the search function is retrieving values having same names	1.Click on the search box 2. Enter a value such that the result has 2 or more results with same name	Test data , Two or more persons should have same name	Name of person		All the values with same name should be displayed					
		STC_006	Enter a name of the person which is not present in the database	1.Click on the search field 2. Enter a Name which is not present in the database	Test data,	Name of person whiich is not present in the database		NO search result should be displayed					
		STC_007	Checking if the Search is case Sensitive	1.Click on the Search field 2.Enter the Name of the person in capital letters	Test data, A Person name having all small letters in the database.	Name of the person having all small letters		The person name with the small letters should be displayed					
		STC_008	Checking if the Search is case Sensitive	1.Click on the Search field 2.Enter the Name of the person in capital letters	Test data, A Person which includes capital letters in the database.	Name of the person which includes Capital letters		The person name with the Capital letters should be displayed					
		STC_009	Checking if we can delete the values entered in the search field	1.Click on the search field 2.Enter a string into the field 3.Delete the Entered String	Test data	String value		Should be able to delete the string entered in the Search field					
		STC_010	Searching with No values	1.Click on the Search field	Test data	No input		Display shouldnot change					
		STC_011	Verify response time of search	1.Click on the Search box, 2.Enter a valid person name present in the database	Test data	Valid name		The search results should be quick					
		STC_012	Verifing if the Blank spaces are considered in the search	1.Click on the search field 2.Enter a valid name which includes space between first and last name	Test data	Valid name which includes a Space in it		The Peron with the Entered name should be displayed					
		STC_013	Verifing if we can select the Name from the Result after completing a successful search	1.Click on the search field 2.Enter a valid name 3.Select the name from the result displayed	Test data	Valid name		Should display the allocations of the selected person					

Test Scenario ID	Test Scenario Description	Test Case ID	Test Case Description	Test Steps	Preconditions	Test Data	Post Conditions	Expected Result	Actual Result	Status	Executed By	Executed Date	Comments
N_001	Items in the Navbar	NT_001	Verifing if all the items in the Navbar are displayed	Check if all the items in the Navbar are displayed	Valid Url			All the expected items in the Nav bar must be displayed					
N_002	Nav links	NT_002	Verifing if the People table is displayed when the people tab is clicked	1.click on the People tab in the Nav bar	Valid Url			People table must be displayed					
		NT_003	Verifing if the Projects table is displayed when the Projects tab is clicked	1.click on the Projects tab in the Nav bar	Valid Url			Projects table must be displayed					
		NT_004	Verifing if the Allocations page is displayed when the Allocationstab is clicked	1.click on the Allocations tab in the Nav bar	Valid Url			Allocations must be displayed					
		NT_005	Verifing if the Spending table is displayed when the Spending tab is clicked	1.click on the Spending tab in the Nav bar	Valid Url			Spending table must be displayed					
		NT_006	Verifing if the Remaining table is displayed when the Remaining tab is clicked	1.click on the Remaining tab the Nav bar	Valid Url			Remaining table must be displayed					
		NT_007	Verifing if the End Balance table is displayed when the End Balance tab is clicked	1.click on the End Balance tab in the Nav bar	Valid Url			End Balance table must be displayed					
		NT_008	Verifing if user can switch between the tabs in the Nav bar	1.Click on any of the Tab from the Nav bar 2.After the expected page is displayed select another tab from the Nav bar	Valid Url			The corresponding page to the second tab that is selected must be displayed					
		NT_009	Verifing that there is no change in the display when the same tab from the Navbar is selected more than once	1.Click on any of the Tab from the Nav bar 2.After the expected page is displayed select the same tab from the Nav bar.	Valid Url			There must be no changes in the display					
		NT_010	verifing the ability of the dropdown display of the Add tab in the Nav bar	1.Click on the Add tab in the Nav bar	Valid Url			The dropdown list of the Add tab must be displayed					
		NT_011	Verifing if the add Staff Page is displayed	1.Click on the Add tab in the Nav bar 2.Click on the Staff tab from the dropdown list	Valid Url			The page providing the add Staff functionality should be displayed					
		NT_012	Verifing if the add Project Page is displayed	1.Click on the Add tab in the Nav bar 2.Click on the Staff tab from the dropdown list	Valid Url			The page providing the add Projectfunctionality should be displayed					

Test Scenario ID	Test Scenario Description	Test Case ID	Test Case Description	Test Steps	Preconditions	Test Data	Post Conditions	Expected Result	Actual Result	Status	Executed By	Executed Date	Comments
T_001	Spending	TT_001	Check if all the items in the Spending table are displayed correctly	1.Click on the Spending tab from the Nav bar 2.The spending table is displayed	Valid url			All the items of the spending table that are present in the database must be displayed					
T_002	Remaining	TT_002	Check if all the items in the Remaining table are displayed correctly	1.Click on the Remaining tab from the Nav bar 2.The spending table is displayed	Valid url			All the items of the Remaining table that are present in the database must be displayed					
T_003	End Balance	TT_003	Check if all the items in the Spending table are displayed correctly	1.Click on the End Balance tab from the Nav bar 2.The spending table is displayed	Valid url			All the items of the Project table that are present in the database must be displayed					

Test Scenario ID	Test Scenario Description	Test Case ID	Test Case Description	Test Steps	Preconditions	Test Data	Post Conditions	Expected Result	Actual Result	Status	Executed By	Executed Date	Comments
P_001	Add new Project to project table	PTC_001	Enter valid entries	1. press add button 2. enter valid values in the fields 3. press save button	Test Data Valid URL	5 samples of Random valid values	message "Added Successfully "	popup "Added successfully"					
		PTC_002	Enter invalid enteries	1. press add button 2. enter invalid values for each field 3. press save button	Test Data Valid URL	5 samples of Random invalid values 5 samples of invalid values	invalid inputs message	popup "invalid inputs"					
		PTC_003	Enter invalid enteries	1. press add button 2. enter invalid values for one or two fields 3. press save button	Test Data Valid URL		invalid inputs message	popup "invalid inputs"					
		PTC_004	No Entries	1. press add button 2. No values Entered 3. press save button	Test Data Valid URL	No entries	No inputs message	popup "No inputs"					
		PTC_005	Enter special charatcers	1. press add button 2. enter special characters for each field 3. press save button	Test Data Valid URL	5 samples of input including special characters	invalid inputs message	popup "invalid inputs"					
		PTC_006	Verifing that the Name field in the table doesnot accept invalid values	1.Click on add Project button 2.ENter valid inputs in the fields except for the Name field 3.Click on the Submit button	Test Data Valid URL	Nine valid and one invalid entries	invalid inputs message	popup "invalid inputs"					
		PTC_007	Verifing that the End date field in the table doesnot accept invalid values	1.Click on add Project button 2.ENter valid inputs in the fields except for the End date field 3.Click on the Submit button	Test Data Valid URL	Nine valid and one invalid entries	invalid inputs message	popup "invalid inputs"					
		PTC_008	Verifing that the Ext. Salary field in the table doesnot accept invalid values	1.Click on add Project button 2.ENter valid inputs in the fields except for the Ext. Salary field 3.Click on the Submit button	Test Data Valid URL	Nine valid and one invalid entries	invalid inputs message	popup "invalid inputs"					
		PTC_009	Verifing that the Ext. Overhead field in the table doesnot accept invalid values	1.Click on add Project button 2.ENter valid inputs in the fields except for the Ext. Overhead field 3.Click on the Submit button	Test Data Valid URL	Nine valid and one invalid entries	invalid inputs message	popup "invalid inputs"					
		PTC_010	Verifing that the Int. Salary field in the table doesnot accept invalid values	1.Click on add Project button 2.ENter valid inputs in the fields except for the Int. Salary field 3.Click on the Submit button	Test Data Valid URL	Nine valid and one invalid entries	invalid inputs message	popup "invalid inputs"					
		PTC_011	Verifing that the Int. Othercost field in the table doesnot accept invalid values	1.Click on add Project button 2.ENter valid inputs in the fields except for the Int. Othercost field 3.Click on the Submit button	Test Data Valid URL	Nine valid and one invalid entries	invalid inputs message	popup "invalid inputs"					
		PTC_012	Verifing that the Ext. STC field in the table doesnot accept invalid values	1.Click on add Project button 2.ENter valid inputs in the fields except for the STC field 3.Click on the Submit button	Test Data Valid URL	Nine valid and one invalid entries	invalid inputs message	popup "invalid inputs"					
		PTC_013	Verifing that the Ext. Other costs field in the table doesnot accept invalid values	1.Click on add Project button 2.ENter valid inputs in the fields except for the Ext. Other costsfield 3.Click on the Submit button	Test Data Valid URL	Nine valid and one invalid entries	invalid inputs message	popup "invalid inputs"					
		PTC_014	Verifing that the Int. Overhead field in the table doesnot accept invalid values	1.Click on add Project button 2.ENter valid inputs in the fields except for the Int. Overhead field 3.Click on the Submit button	Test Data Valid URL	Nine valid and one invalid entries	invalid inputs message	popup "invalid inputs"					
ST_002	Modifing the data of a particular project in the Project table Table	PTC_015	Selecting a Project in the Staff Table	1.Click on the Project row which needs to be selected	Test Data Valid URL			Should be able to select the Project row					
		PTC_016	Modifing the present data in the Project Table with valid data	1.Click on the field to modify 2.Delete the existing data 3.Enter valid data 4.Save the changes	Test Data Valid URL	valid data		Should be able to modify the data present in that field					
		PTC_017	Modifing the present data in the Project Table with invalid data	1.Click on the field to modify 2.Delete the existing data 3.Enter invalid data 4.Save the changes	Test Data Valid URL	invalid data		popup "invalid inputs"					
		PTC_018	Modifing the present data in the Project Table with Null data	1.Click on the field to modify 2.Delete the existing data 3.Save the changes	Valid Url	No entries		popup "invalid inputs"					
ST_003	Saving the data	PTC_019	Verifing that the data is saved in the database	1. press add button 2. enter valid values in the fields 3. press save button 4. close and reopen the application. 5.Click on the Project table from the Nav bar	Test Data Valid URL	valid data		The data should be saved correctly in the database					
		PTC_020	Verifing that the Modified data is saved in the database	1. press add button 2. Modify the the data in the table with valid data 3. press save button 4. close and reopen the application. 5.Click on the People table from the Nav bar	Test Data Valid URL	valid data		The data should be saved correctly in the database					

Test Scenario ID	Test Scenario Description	Test Case ID	Test Case Description	Test Steps	Preconditions	Test Data	Post Conditions	Expected Result	Actual Result	Status	Executed By	Executed Date	Co
A_001	Display Information	AT_001	Verify whether all the items that must be shown are dispplayed on the Allocation page	1.click on the Allocation tab from the Nav bar				All the items must displayed on the page					
			User must be able to view the allocations of a user by selecting on the name from the list	1.click on the Allocation tab from the Nav bar 2.Click on any of the Person from the list				All the allocations and the projects of the selected person must be displayed					
			There must be no change in the display when the same person is selected twice	1.click on the Allocation tab from the Nav bar 2.Click on any of the Person from the list 3.click on the same person again				There must be no change in the display					
			User must be able to switch the display view from one perosn to another	1.click on the Allocation tab from the Nav bar 2.Click on any of the Person from the list 3.click on another person from the list				The details of the last selected person must be displayed					
A_002	Timeline	AT_002	Verify whether zooming in on the timeline would give more detailed view of the calender	1.click on the Allocation tab from the Nav bar 2.Place the Mouse cursor on the timeline 3.Zoomin on the timeline				The time line must display more detailed view of the calender					
		AT_003	Verify whether zooming in on the timeline would give more detaled view of the calender	1.click on the Allocation tab from the Nav bar 2.Place the Mouse cursor on the timeline 3.Zoom out on the timeline				The time line must display less detailed view of the calender					
		AT_004	Verify whether user can scroll the timeline from left to right or vice versa	1.click on the Allocation tab from the Nav bar 2.Place the Mouse cursor on the timeline 3.scroll to the right and left				user must be abke to scroll through the timeline					
A_003	Allocation bar	AT_005	Verify if the user can create a new allocation bar on the timeline	1.Click on the allocation tab from the Nav bar 2.Double click on the time timeline				New allocation must be created					
		AT_006	Verify if the user can add the data on to the allocation bar	1.Click on the allocation tab from the Nav bar 2.Double click on the time timeline 3.new allocation bar is created. 4.double click on the allocation bar 5. Enter a valid Float value		Float value		User must be able to view the entered value on that particular allocation bar					
		AT_007	User cannot add invalid data into the allocation bar	1.Click on the allocation tab from the Nav bar 2.Double click on the time timeline 3.new allocation bar is created. 4.double click on the allocation bar 5. Enter the invalid value		5 samples of invalid values		popup "invalid data"					
		AT_008	Verify if the user can delet the data entered into the allocation bar	1.Click on the allocation tab from the Nav bar 2.Double click on the time timeline 3.new allocation bar is created. 4.double click on the allocation bar 5. Enter a valid value 6.save 7.Delete the entered value		valid value		User must be able to delete the value from the allocation bar					
		AT_009	Verify if the user can modify the data entered into the allocation bar	1.Click on the allocation tab from the Nav bar 2.Double click on the time timeline 3.new allocation bar is created. 4.double click on the allocation bar 5. Enter a valid value 6.save 7.Modify the entered value		valid value		User must be able to Modify the value from the allocation bar					
		AT_010	The Allocation bar cannot overlap on each other	1.Click on the allocation tab from the Nav bar 2.create two allocation bars on th time line 3. drag and place one allocation bar on the other				pop up "Allocation bars cannot overlap"					
		AT_011	User must able to drag the allocation bar horixontally to increase the peroid of allocation	1.Click on the allocation tab from the Nav bar 2.create an allocation bars on th time line 3. stretch the allocation bar horizontally				User mjust be able to strtch the allocation bar by dragging it horizontally					
		AT_012	Verify if the user is able to delete the allocation bar	1.Click on the allocation tab from the Nav bar 2.create an allocation bars on thetime line 3. delte the allocation bar				The allocation bar must be deleted					
A_004	Total	AT_013	Verify if the total percentage is displayed correctly for any given of time	1.Click on the allocation tab from the Nav bar 2.create allocation bars on different projects, 3.Enter valid values in the allocation bar 4. check if the total is shown correctly at any point of time in the timeline				The total must be correct at any point of time					
		AT_014	The total percentage of the total must not be greater an 100 at any given point on the timeline	1.Click on the allocation tab from the Nav bar 2.create allocation bars on different projects, 3.Enter valid values in the allocation bar such the total shoukd be greater than 100				popup message "The total cannot be greater than 100"					
		AT_015	Modification in the allocation bar must be reflected on the total timeline as well	1.Click on the allocation tab from the Nav bar 2.create allocation bars on different projects, 3.Enter valid values in the allocation bar 4. check if the total is shown correctly at any point of time in the timeline 5.Modify the allocation bars				The changes done on the allocation bars must be reflected on the total timeline					
A_005	Toggle	AT_016	When toggle button is clicked only the projects with allocations for the selected person must be displayed	1.selecta person from the list 2.create allocations for some of the projects in the list 3.click on toggle button				Only the projects having allocations should be displayed					
		AT_017	Clicking the toggle twice should display all the projects	1.selecta person from the list 2.create allocations for some of the projects in the list 3.click on toggle button 4.check if only the projects with the allocations are displayed 5.click on the toggle button				All the projects present must be displayed					
A_006	Database	AT_018	The data from the database must be displayed correctly	1.Click on the Allocation tab from the Nav bar							Verify that the data displayed is correct when compared to the data in the database		
		AT_019	Creating a new allocation and saving it	1.Click on the Allocation tab from the Nav bar 2. select a perosn from the list 3.Create a allocation for that person on any of the project 4.Save it 5.Close and Re-open the application 6.Click on the allocation tab from the Nav bar 7.Select the same person that was selected previously							The new allocation created must be displayed		
		AT_020	Modifing the allocation and saving the changes	1.Select a person from the list 2.Create an allocation for that person 3.save it 4. reopen the application 5.select the same person and modify the allocation 6.save and Re=open the application							The modifications made to the allocations should be displayed		
		AT_21	Deleting the allocation from the database	1.Select a person from the list 2.Create an allocation for that person 3.save it 4. reopen the application 5.select the same person and delete the allocation 6.save and Re=open the application							The alllocation deletd should not be displayed		