Software development report

Group report

Gisylia Geyoro, André Brasil, Kevin Blandford, Anton | engineering software systems | May 15, 2019

Contents

[Team roles and responsibilities 2](#_Toc8848991)

[Focus Questions 3](#_Toc8848992)

[Risk analysis table 4](#_Toc8848993)

[System development methodology Justification 5](#_Toc8848994)

[Requirements documentation 5](#_Toc8848995)

[Use case texts 7](#_Toc8848996)

[User interface 13](#_Toc8848997)

[Design plan 13](#_Toc8848998)

[Tools and communication means 14](#_Toc8848999)

# Team roles and responsibilities

Using our Belbin self-prediction test and our individual concept maps, we can decide to allocate each task based of our strengths and weaknesses as well as selected jobs we felt most comfortable with.

Separating all the tasks into 3 main milestones we chose our own tasks to complete making this more comforting. We decided to do this method as it made completing tasks quicker

Milestone 1 – Task allocation

|  |  |  |
| --- | --- | --- |
| Task | Member | Justification |
| Focus questions | Team |  |
| Gant Chart | Gisylia | Good organisational skills. |
| Risk Analysis | Team |  |
| Work Breakdown Structure (WBS) | Team |  |
| Access levels | Team | Multiple perspectives on the project. Allowed everyone to fully understand the project. |
| Project Concept Map | Andre, Antons | Everyone else was busy. |
| Team Concept Map | Antons | Antons had the most experience with concept maps. |

Milestone 2

|  |  |  |
| --- | --- | --- |
| Task | Member | Justification |
| Use case and use case texts | Andre | Equally distributed tasks based on preferences |
| Domain model | Antons |  |
| Robustness diagram | Kevin |  |
| Sequence diagram | Gisylia |  |
| Class diagram | Antons |  |
| Interface design | Gisylia | Had to wait for everyone else’s work |

Milestone 3

|  |  |  |
| --- | --- | --- |
| Task | Member | Justification |
| JSPs | Gisylia, Kevin | we had decided that we would take tasks based on how comfortable we felt in each area. |
| Gluon GUI | Gisylia |
| Web server | Antons |
| Database | Antons |
| Classes | Andre |

# Focus Questions

**What is the main use of the system?**

The system is being developed for a school to better the school’s current system of information teacher, parents and pupils of the different activities that happen during school times. The system also must provide a means of online payment making it faster and easier to do.

**Who is the system helping?**

The system will be designed to assist the teachers, pupils, parents and the schools board

**How does a user use the system?**

The users will already have set username and password done by the admin in the database, which they can then use to login and conduct various actions depending on their access level.

**What are the systems demographic user base?**

The system can be used and accessed by anyone who has an account and is registered to the system. There are no age restrictions and follow the GDPR act as well as obey the Malicious Communications Act

**When can the system be used?**

As it is a web-based system. The user can access it anytime at any location if there is a means connectivity.

**Why should this system be used instead of an alternative solution?**

The system will be designed to make doing certain actions which originally would have to be done over a long period of time in a shorter span. It is also an easy way to gain access to students’ information quicker and on the move.

**What technology will be used to create the system?**

PHP, MYSQL, HTML, CSS, JS, JavaFX, Gluon

**Will the system require any new technology to make it work?**

No

**What requirements are needed for it to function?**

The application and connectivity

**will the system differ on a different device, such as a mobile?**

The system will be able to fully function on different devices

**Who will maintain the system?**

The systems admins

**Will the system be available in the future?**

yes.

# Risk analysis table

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Potential Risk | Impact rating | Probability rating | Risk score | Possible solutions | Trigger |
| Team risks | | | | | |
| Team member doesn’t follow the agreed plan | High | Low | 2 | Give them a strike, report them to Femi if they exceed strike limit | Team members opinions are different compared to everyone else’s |
| Issues using new technologies | Low | High | 3 | Research, or find a different solution | Team members are unaware of new languages or technology |
| Not completing work by deadline | High | Low | 2 | Efficient time management | Team members miss due dates or take longer than expected |
| Team member struggling to keep up with the rest of the team | Low | High | 3 | Spend more time working on a solution, get help from more experienced team members | Team members are unaware of how to complete the task set |
| Team members are not getting along | Low | Low | 4 | Reallocate tasks to different team members | An issue occurs within the group and it isn’t resolved quickly |
| System risks | | | | | |
| Client risk - The potential change of clients’ needs | High | Low | 4 | Ensure that all the information collected by the client is always up to date and ensure that there is a method in place to conduct changes if necessary | The client decides to change some specifications |
| Design- the designs are different to what the customer wanted | High | Medium/low | 3 | Ensure that the team member understands what the customer want and keeps UpToDate with each change | Client isn’t happy with the design choices |
| Implementation- the system doesn’t work | High | Medium/High | 2 | Ensure that each team member is UpToDate and isn’t having any issues completing each task | Team members struggle to code |
| Testing – unable to do any testing | High | Low | 4 | Ensure that the team is on task and there is an enough time before testing | The team is behind, and the system isn’t finished |
| Maintenance- the system is unable to be maintained after completion | High | High | 1 | Implement preventative/perfective maintenance to ensure the detection of any latent faults in the software and correct them before they become effective faults | There wasn’t enough testing conducted |

# System development methodology Justification

Looking at the list of different methodology’s and comparing them, we had decided that the best method that would fit in with our development plan was the Spiral methodology. This being because the method fits our project requirements, is beginner friendly as we are using unfamiliar technology, is very simple and easy to follow.

# Requirements documentation

Functional

1. Must have:

* Access levels
* Database
* Different classes for each access level
* Activities
* Image/text upload
* Payments/receipts
* Multiple schools
* Login system
* Associated guardian and tutor and school to pupils

1. Pupil

* Can view activities
* Upload files for activities

1. Teachers

* Can manage free activities (create, modify, delete, view)
* Can manage pupil
* Can do everything that pupil can
* Can link pupil account to parent account

1. Parents

* Can make payments
* Can refund payments
* Can do everything that pupil can
* Can manage (view, create, modify, delete only ones that were created by them) multiple pupil activities

1. Board

* Can manage teachers
* Can manage parents
* Can manage invoices
* Can manage paid activities
* Can do everything that teacher and parents do
* Can manage multiples of above activities

1. Admin

* Create, remove, modify schools(board)
* And do everything that board does

1. Activity

* Can be group
* Can be individual
* Should be linked to user or group

1. Payments

* Can only be made by parents
* Can be refunded

Non-functional

1. Should have:

* Good design
* Security
* Performance

# Use case texts

ADMIN USE CASES

Creating a new school

|  |  |  |
| --- | --- | --- |
| STEPS | USER ACTION | SYSTEM RESPONSE |
| 1 | Use case starts when ADMIN creates a school |  |
| 2 |  | System ask for school details |
| 3 | User enters school details |  |
| 4 |  | System adds school to system |
| 5 |  | System redirects user to initial page |
| 6 |  | System shows new school on dashboard |

Creating a parent

|  |  |  |
| --- | --- | --- |
| STEPS | USER ACTION | SYSTEM RESPONSE |
| 1 | Use case starts when ADMIN selects create parent option |  |
| 2 |  | System prompts parent details |
| 3 | User enters parent details |  |
| 4 |  | System checks if there are any matches. If so, system shows notification of existing data. Redirects user to step 1. |
| 5 |  | System adds parent |
| 6 |  | System redirects user to step 1. |

Displaying school options

|  |  |  |
| --- | --- | --- |
| STEPS | USER ACTION | SYSTEM RESPONSE |
| 1 | Use case starts when ADMIN select display schools’ option |  |
| 2 |  | System presents a list of schools |
| 3 | User clicks the desired school |  |
| 4 |  | System gives a list of options between view teachers, view pupils and view parents |
| 5 | User selects view teachers |  |
| 6 |  | System shows teachers list with delete or create option |
| 7 | User selects create option |  |
| 8 |  | System prompts teacher details |
| 9 | User enters details |  |
| 10 |  | System checks for any matches |
| 11 |  | If there are no matches. System displays successful notification and redirects user to main screen |

Creating a new activity

|  |  |  |
| --- | --- | --- |
| STEPS | USER ACTION | SYSTEM RESPONSE |
| 1 | Use case starts when ADMIN selects create activity option |  |
| 2 |  | System ask for activity details |
| 3 | User enters school details |  |
| 4 |  | System adds activity to system |
| 5 |  | System displays new activity on dashboard |
|  |  |  |

**SCHOOL BOARD**

Creating an invoice

|  |  |  |
| --- | --- | --- |
| STEPS | USER ACTION | SYSTEM RESPONSE |
| 1 | Use case starts when BOARD selects create invoice option |  |
| 2 |  | System prompts invoice details (dates, prices, information etc) |
| 3 | User enters invoice details |  |
| 4 |  | System adds invoice |
| 5 |  | System displays new invoice on parents account |
|  |  | User gets redirected to main menu |

Deleting invoices

|  |  |  |
| --- | --- | --- |
| STEPS | USER ACTION | SYSTEM RESPONSE |
| 1 | Use case starts when ADMIN selects invoice list option |  |
| 2 |  | System displays list of invoices |
| 3 | User selects invoice to delete |  |
| 4 |  | System deletes invoice |
| 5 |  | System displays new activity on dashboard |
|  |  |  |

**TEACHER**

Creating a new activity

|  |  |  |
| --- | --- | --- |
| S TEPS | USER ACTION | SYSTEM RESPONSE |
| 1 | Use case starts when TEACHER selects create activity option |  |
| 2 |  | System ask for activity details |
| 3 | User enters activity details |  |
| 4 |  | System adds activity to system |
| 5 |  | System displays new activity |
|  |  |  |

Adding pupil

|  |  |  |
| --- | --- | --- |
| STEPS | USER ACTION | SYSTEM RESPONSE |
| 1 | Use case starts when parents click add pupil option |  |
| 2 |  | System ask for pupil details |
| 3 | User enters pupil details |  |
| 4 |  | System prompts the user for parent’s identification |
| 5 | User enters parent identification |  |
| 6 |  | System links pupil with parent |
| 7 |  | Parent account has now pupil added |

**PARENT**

Login

|  |  |  |
| --- | --- | --- |
| STEPS | USER ACTION | SYSTEM RESPONSE |
| 1 | Use case starts when PARENT is logs into their account |  |
| 2 |  | Systems presents a list of pupils related to the parent |
| 3 | User selects pupil |  |
| 4 |  | System displays two options:  Check invoices, view pupil’s activities |
| 5 | Parent selects pupil’s activities option |  |
|  |  | System displays pupils’ activities |

Listing pupil and invoice

|  |  |  |
| --- | --- | --- |
| STEPS | USER ACTION | SYSTEM RESPONSE |
| 1 | Use case starts when PARENT logs into their account |  |
| 2 |  | System presents a list of pupils linked to parent’s account |
| 3 | User selects pupil |  |
| 4 |  | System displays two options:  Check invoices or view pupil’s activities |
| 5 | User selects check invoices |  |
| 6 |  | System displays invoices with a payment option |
| 7 | User clicks payment option on desired invoice |  |
|  |  | System redirects user to payment page |

**PUPIL**

Login and uploading files

|  |  |  |
| --- | --- | --- |
| STEPS | USER ACTION | SYSTEM RESPONSE |
| 1 | Use case starts when PUPIL logs in to their account |  |
| 2 |  | System displays a page feed with all activities and upload files option |
| 3 | Pupil selects upload files |  |
| 4 |  | Systems redirects user to an upload window |
| 5 | User selects files to upload |  |
| 6 |  | System checks for valid files (limited file size). If the file is oversized, user gets notification error |

# User interface

### Design plan

Login page

* As user will already have a pre-created login account there will only be the option to login.
* There will be two sides to this page.
  + On the left there is a sign up a new school, where the user can select to add a new school if they are a board member along with a short explanation asking the user if they are a new school.
  + On the right is the option to login with an already set username password and school ID
    - The colour scheme is on this page is a light blue and white.
    - Maintaining consistency is very important so the same blue will be a reoccurring factor in the application
  + To minimize clutter on the page only what is needed will be displayed on the page.

Registering a new school

* Keeping the white and blue colour scheme the register page be in a descending order making it easier for the user to maneuver across the screen.
* The contrast of colour’s isn’t too harsh on the eyes.

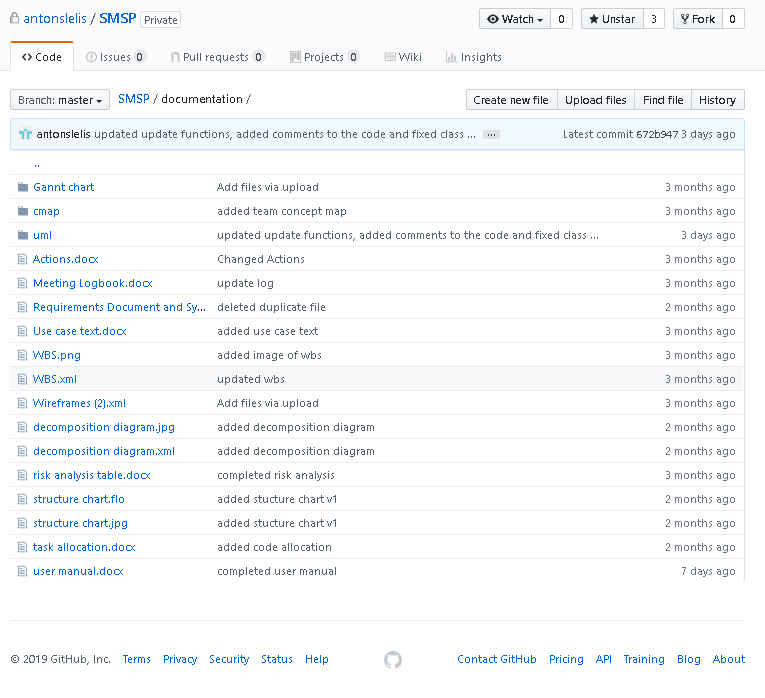
Systems home page

* The user will automatically be redirected to their own home page.
* The home page with be design with hierarchy as its focus but also prioritize navigability, consistency and simplicity.
* The navigation bar will now be visible for quick actions
  + On the top right corner, the user’s information will be displayed depending and depending on the account type, either parent, student, teacher or board will show.
  + Top left corner the users school name will display accompanied with an events, report and payment tabs

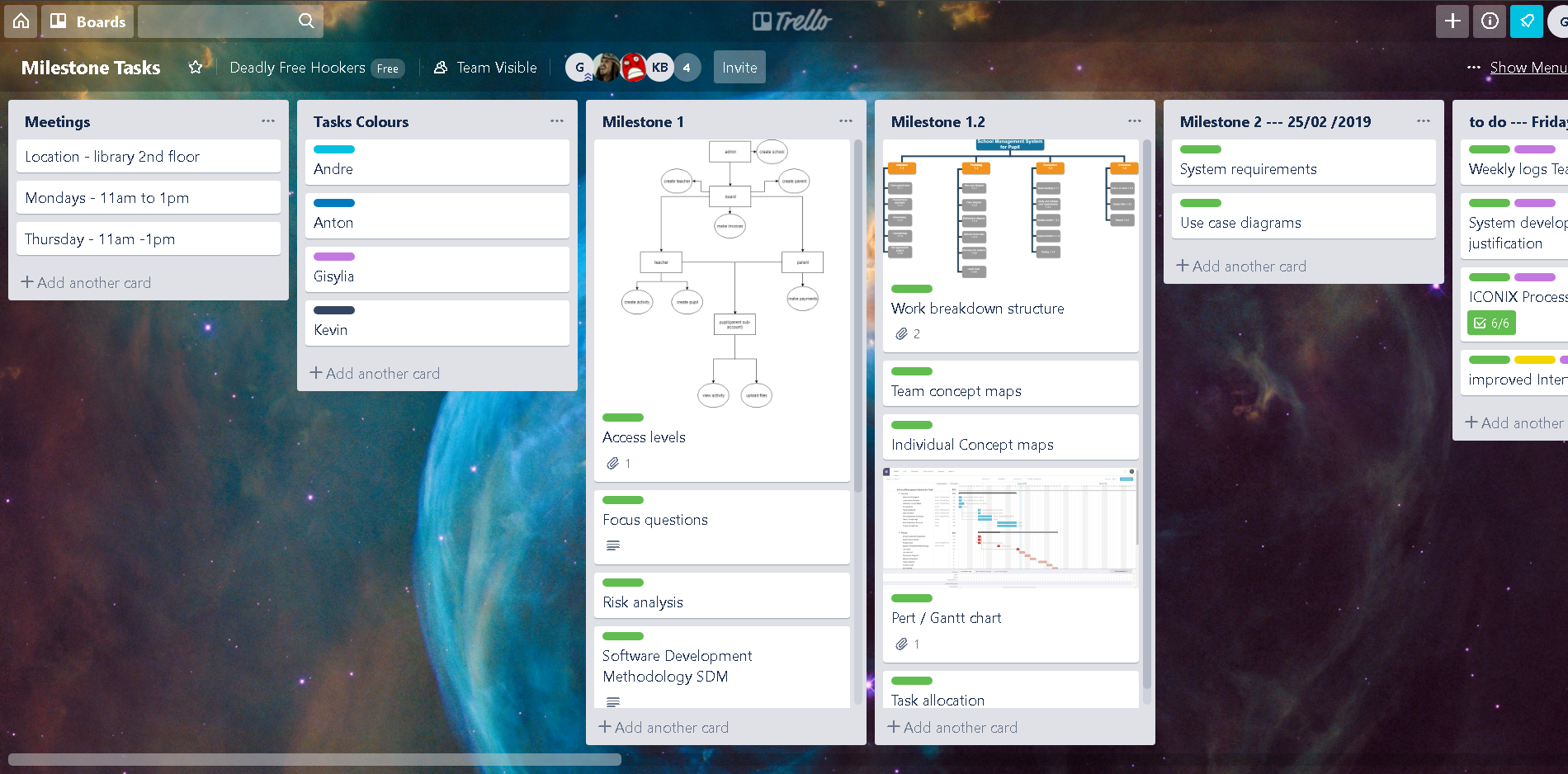
# Tools and communication mean

* WhatsApp was used as our primary means of communication
* Library every Monday at 11am was our default meeting times
* Class times was used as a milestone for the week

Git hub was used to upload, share and store out work processes between us



Trello was used to view the different task each user had to complete. This was done by assigning different colour to each member



Microsoft OneNote was used to keep track of the teams progress each week.

