Group 97 D1 Design Report

Group members: Maksym Yahnyshchak, Rhys Jones, Hoi Hei Ng, Razzaq Rumaan, Nik Amzar Syahmi

Table of Contents

1. Introduction to our Vision for the Software 2
2. Software Requirements 2
3. Desired features 3
4. UI mock-up 4
5. Code design 4
   5.1 Data Structure: 4
   5.2 Ownership: 5
   5.3 Code Modules: 5
6. Design Principles 7
7. Project Plan 8
8. Risk Analysis 10
9. Costing 11
10. Acceptance Tests 14
11. Task List 17
12. Activity Network 18
   12.1 Gantt Chart 18
   12.2 Critical Path Diagram 20
   12.3 Activity Network Diagram 20
13. Appendix 20
   13.1 UI 20
   13.2 UML Class Diagram 22

1. Introduction to our Vision for the Software

For our software, we plan to create a presentation program called SlideCraft which develops the best attributes from other programs, including essential features of well-known programs and interesting ideas from smaller ones. Additionally, we will be introducing new features which specialize our software for usage in computer science modules as a tool for education. For example, a simple code runner in which code can be run during a presentation to enhance computer science lectures with live demonstrations. We hope to implement a striking design for our user interface which will be simple to use whilst providing in-depth features for users

2. Software Requirements

Functional Requirements

1. The users shall be able to open, delete, create, edit and save files.
2. The software shall allow users to add and remove slides.
3. The software shall allow users to manage the slides. This includes copy and paste, duplicate, re-order slides, zoom in and out and scroll through the slides.
4. The software shall allow users to add and remove text elements into the slides in the form of a text box.
5. The software shall allow users to format the text elements. This includes resizing the text box, choosing the type and size of fonts, colors, alignments, indentation, underline, bold, and italic.
6. The users shall be able to add and remove lists, bulleted, and numbered lists into slides.
7. The software shall allow users to insert and remove shapes into the slides.
8. The software shall allow users to insert and remove images and videos into the slides.
9. The software shall allow users to insert and remove hyperlinks into the slides.
10. The software shall allow users to use basic keyboard shortcuts such as copy, cut, paste, save, undo, and redo.
11. The software shall allow users to insert tables, charts and graphs into the slides.
12. The software shall allow users to format all the insertable elements.
13. The users shall be able to add speaker notes into the slides.
14. Software shall provide templates, layouts, transitions, animations options.
15. Software shall provide functionality to preview and present slides.

Non-functional Requirements

1. Software shall support slides with at least 20 pages and complex contents.

2. The users shall understand how the whole program works within 5 mins. Keeping our software simple, avoiding any confusion caused by the users.
3. Software shall store progress that users have been made automatically every 30s. This allows users to not need to save their files manually.
4. Software shall support users with disabilities or different levels of expertise
5. Response time for the software shall be maintained within 1-2s.

3. Desired features

UI features:

- Main toolbar holding tabs, users can choose what they mainly want to do on slides.
- Ribbon layout holding detail commands for the users to choose.
- Slider for viewing slides outline and selecting specific page on the slide.

Functionality features:

- Users should be able to screen record slides as a video.
- Users should be able to manipulate RGB palette to customize colour.
- Users should be able to search for keywords on slides.
- Save editor working progress, prevent users do not save the files.
- Comments should be able to be left by the Users, which allows developers able to fix any issue related to it.

Accessibility & inclusivity features:

- Typing spell check for users.
- Customize keyboard shortcuts to make it comfortable for all users.
- Switch between editing, viewing and colorblind mode for users who need access.
- Tutorials for new users get started with the software.

User engagement features:

- Users should be able to insert some extra components into slides.
- Users should be able to draw when presenting.
- Users should be able to write code in the application and run it when presenting.

4. UI mock-up

This UI sketch represents how our product is expected to look. It will have a taskbar on the left. User can select section and relevant submenu will pop up like in the example above. Our 'Home' menu will have lots of different features such as font style, font size, RGB color pallet, text alignment, etc. In the top left corner will be an option to change file name or mode (Edit/View/Colourblind). In the bottom left corner, we can see such features as: undo, redo, save, present, settings and speaker notes. In the top right corner will be a search box with the option to switch to regex. Also, there is the option to add slides by clicking on the plus sign. Slide panel on the right will support scrolling, removing, reordering and other actions regarding the slides. Users can use zoom feature in the bottom right corner. Also, additional features such as code editor and spell checker can be seen in the toolbar as "#" and "T" icons respectively. UI mockup of insert section can be found in the appendix of this document.

5. Code design

5.1 Data Structure:

- Class Presentation {
  private List<Slides> slideList;

```
        private String name;

        public void addSlide(Slide slide) {
            slideList.add(slide);
        }
    }
```

- ```
  Class Slide {
      private List<SlideObject> slideObjectsList;

      public void addSlideObject(SlideObject slideObject) {
          slideObjectsList.add(slideObject);
      }
  }
  ```
- ```
  Class SlideObject {
      private int x;
      private int y;
      private int height;
      private int weight;
  }
  ```
- ```
  Class TextBox extends SlideObject() {
      private String text;
  }
  ```
- ```
  Class Photo extends SlideObject() {
      private String filename;
  }
  ```
- ```
  Class Shape extends SlideObject() {
      private enum type;
      private String color;
  }
  ```

5.2 Ownership:

- The presentation class contains a list of slides which contain information about what will be presented, Slide class contains a list of elements that are present on the specific slide.
- The Presentation class stores a List of Slides and the name of the presentation. The class SlideObject represents any object that can be added to a Slide, ranging from a photo to a shape such as a square, or even a textbox. For each SlideObject, we need to store the x and y values of the object to be able to store where the objects are on the Slide. We would also need to store the width and height of each object. For an Osuch as a shape, we would need to store the color of the shape as well.

5.3 Code Modules:

- Our main code will be contained inside the src/main directory. We will include tests inside the src/test directory. The directory src/main also contains a folder named resources, which contains any resources we will use such as images or svg files.
- Inside the main directory, we will contain the following modules:
  - UI Module – includes ActionListener and KeyListeners.
  - Data Module – contains classes in which we model our presentations around. These classes include Presentation, Slide, SlideObject, etc.
  - File I/O Module – classes for the importing or exporting (or saving) of the presentation.

We have a UML class diagram shown in Appendix 13.3.
  ```

Below is a component diagram, which shows the relationship between different components in a system. The Application class is a service provider for the UserInterface class, which is a service user that will call operations to update the UI. The Application class can interact with the database by saving or requesting data from the database. Application class will utilize the functionality defined in the classes ending in Manager to provide a service to the UI – example of the functionality provided is:

- FileManager which is used for the opening and saving of files.
- InsertManager which is used to add a new element to the slide – this element can be of different types such as images, hyperlinks, tables, etc.
- DesignManager contains templates and transitions used to improve the slides design.
- ElementsManager which is used to select elements, change the color of the elements, and copy/paste elements.

6. Design Principles

An important design principle to consider is DRY or Don't Repeat Yourself. This principle ensures that duplicate code is not written and can be put into practice through using Abstraction using interfaces or abstract classes. This principle helps with the maintenance of the code.

A key group of principles to follow are the SOLID Design Principles:

- Single Responsibility states that a class should only have one specific purpose. This reduces the coupling between the individual component of the software and the code – changing one functionality will not affect any other functionality as it is no longer coupled.
- Open-Closed states that classes should be open for extensions but closed for modifications. This ensures that tested code that is guaranteed to work will be left untouched and will not break.
- Liskov Substitution states that a superclass should not have more functionality than the subclass and that subtypes should be substitutable for a supertype. Essentially, this means that methods or functions which use superclass type must be able to work with the object of subclass without any issue.
- Interface Segregation states that a client should not be able to implement an interface if it does not use it – this occurs when an interface contains more than one functionality and the class that is implementing it only requires a specific functionality.
- Dependency Inversion states that developers should rely on abstractions and not concretions. It is a principle that allows our code to be extensible, loosely coupled and maintainable. One way to perform Dependency Inversion is by creating an interface, and then implementing the functionality described in the interface. In our example, we could create an interface for a type of object such as Shape, and then implement the concrete implementation in classes such as Square or Circle.

By assigning each module a single actor, we will put the Single Responsibility Principle into practice. For example, we will have a distinct class named "Presentation" that will store all of the slides, a class for slide objects, and a class for each slide individually. We will use the Liskov Substitution Principle to determine whether the code makes sense to implement when we create a new class that inherits an existing one.

For our slides, we'll apply the Open Closed Principle. A class called "Slide" (whose name may vary while it's being developed) will accept an interface called "SlideObjects" as an argument. As a result, adding new components doesn't have to worry about breaking already-written code—something that may happen if all the slides are added to the same class.

A Design Principle like Dependency Inversion is favoring Inheritance over Composition. This principle allows for the flexibility to change the functionality with a better implementation at any time.

7. Project Plan

This table showcases a brief overview of our weekly planning on the project. It includes all the weekly tasks that would need to be done to achieve the desired milestones of every 3 weeks in the 11-week schedule. The project plan is designed according to the level of group members coding abilities to ensure its feasibility. After each milestone, we would do an internal review and testing to gather feedback as well as to check any encountered issues which we will bring attention to and put right in the week after, before proceeding with the succeeding tasks.

Week	Task(s)
Weeks 4-6	Brainstorming ideas and discussing plans for project Creating an online channel (Whatsapp) to organize meetups Working on the design report
Week 7	Creating the main window UI Adding and removing slides Implementing file management abilities (open, create, save, close files)
Week 8	Working on the main window UI Add option to change color of background slides Add features to manage the slides (Copy/Paste, Duplicate, Re-order slides, zoom in/out and scroll through slides)
Week 9	Adding and removing text components Copy/Paste Texts Implementing formatting features on text elements (text box size, type & size of fonts, colors, alignments, underline, bold, italic etc.)

Week 9 Milestone By the end of Week 9, we would have a functioning presentation program that could execute the most basic of tasks. The program should have a main UI, where the slides will be created and edited, it should also be able to handle basic file management tasks. The first two main components, slides and texts would be added as well as the ability to modify/format the elements.	
Week 10	Adding list/bullet points including text alignment, spacing, indentation Adding undo/redo feature Adding staple keyboard shortcuts such as copy, paste, and undo/redo
Week 11	Inserting Png/jpg images, videos, shapes, tables, charts and hyperlinks Implementing formatting features on insertable elements
Week 12	Presentation mode viewing option including (slide transitions, sound, exit, shortcuts) Adding option to apply slides layout preset/templates Adding feature that could group the elements Adding search by text feature
Week 12 Milestone By the end of Week 12, we would have a refined code to address feedback/issues from Week 9 review. In addition to that, as well as add a few new functional requirements to further build on its functionality.	
Week 13	Add speaker notes Add feature for slide annotation when presenting
Week 14	Integrate color blind mode Implement Code Runner on the slides

Week 15	Final preparation and testing before submission

Week 15 Milestone

By the end of Week 15 our project would be completed for submission, we would have cleaned up the main code base to be fully functional as required. We would also have made a final decision on which higher risk features to add to the project and implemented as much as possible within the time frame.

8. Risk Analysis

Element	The Risks	Risk Score
Feature Creep	Adding too many features than can reasonably be implemented in the project's timeframe	6/10
Code Runner	Implementing a live code runner in the program could prove to be too complex as nobody in the team has experience doing anything similar so it is hard to estimate the time it would take to do. However, the feature is quite isolated so would not involve many other systems	8/10
Spell Check	Will need to find a suitable library to make use of as coding from scratch would be too large a scope for this project. Crucial to the project/ Would be used in almost every area that uses text	5/10
Adding images/ shapes/capability to draw	Some of this will be available through swing which also provides a way to move objects	4/10
	Will need to find a suitable library to make use of as coding from scratch would be too large a scope for this project	
Adding colorblind modes	Will need to find color schemes that make program accessible for as many people as possible	2/10
Having undo/redo capabilities	Will require multiple versions of a presentation to be stored or to keep a record of every action the user performs to be	6/10

	used as an inverse. Important to have to ensure a responsive program and will likely be used by every part of the system	
Save/Load of files	Requires a file that can store all attributes of a presentation. Important to allow users to leave and return to the application	5/10
Slide animations	Will need a way to add animations and apply them to the program's slides	8/10

We can use these risk scores to assist in our project plan by comparing the importance of each risk to its score to determine how much time to give to developing it. For example, saving and loading files is crucial to the program but also has a high-risk score so should be planned to be done early in development as to ensure that it gets completed. Whereas the code runner has a very high-risk score but is not as important to the program so can be done later in development if there is time to do it.

9. Costing

	Cost per hour (1 person)	**Estimated Hours (all per.)**	**Estimated Sub-total**
Planning and Design phase (w4-6)		56	£1,463.40
Team on-boarding (eg, induction, technical setup)	£15.50	2.5	£38.75
Initial topic selection	£40.00	2.5	£100.00
Feature selection and principles	£40.00	10	£400.00
Requirements analysis	£20.55	5	£102.75
UI design	£40.00	5	£200.00
Architectural design	£20.55	4	£82.20
Risk analysis	£20.55	3	£61.65
Acceptance tests	£15.50	3	£46.50

Schedule planning (tasks, activity & gantt)	£20.55	6	£123.30
Design report writing and editing	£20.55	15	£308.25
Implementation phase (w7-15)		279	£9,620.70
Milestone 1 development (w7-9)		93	£3,206.90
Design (eg, class design)	£20.55	8	£164.40
Development (coding)	£50.00	30	£1,500.00
Documentation	£50.00	10	£500.00
Testing	£15.50	35	£542.50
Coordination (eg, additional meetings, meetings)	£50.00	10	£500.00
Milestone 2 development (w9-12)		93	£3,206.90
Design	£20.55	8	£164.40
Development	£50.00	30	£1,500.00
Documentation	£50.00	10	£500.00
Testing	£15.50	35	£542.50
Coordination (eg, additional meetings, meetings)	£50.00	10	£500.00
Milestone 3 development (w12-15)		93	£3,206.90
Design	£20.55	8	£164.40
Development	£50.00	30	£1,500.00
Documentation	£50.00	10	£500.00
Testing	£15.50	35	£542.50
Coordination (eg, additional meetings, meetings)	£50.00	10	£500.00
User Evaluation phase (w16-17)		163	£5,816.90
Evaluation design	£40.00	10	£400.00
Ethics approval process	£40.00	5	£200.00

Participant recruitment	£40.00	5	£200.00
Running evaluation	£15.50	10	£155.00
Analysis	£50.00	20	£1,000.00
Resulting changes		113	£3,861.90
Design	£20.55	8	£164.40
Development	£50.00	35	£1,750.00
Documentation	£50.00	10	£500.00
Testing	£15.50	45	£697.50
Coordination (eg, additional meetings, meetings)	£50.00	15	£750.00

Non-staff costings	**P&P phase**	**M1 phase**	**M2 phase**	**M3 phase**	**User Eval phase**	**Sub-total by cost**
Developer equipment (eg, computers)	£0.00	£0.00	£0.00	£0.00	£0.00	£0.00
Hosting costs	£0.00	£0.00	£0.00	£0.00	£0.00	£0.00
Software subscriptions	£10.00	£10.00	£10.00	£10.00	£10.00	£50.00
User eval participant costs	£0.00	£0.00	£0.00	£0.00	£400.00	£400.00
Sub-total by phase	£10.00	£10.00	£10.00	£10.00	£410.00	

Total person-hours effort	498
Total staff costs:	£16,901.00
Total non-staff costs:	£450.00
Total project cost estimate:	**£17,351.00**

Planning and Design Phase cost

The total cost for the Planning and Design phase is around £1,463.40. This covers everything from getting the team ready and choosing topics to designing the user interface. It's a careful balance of time and money, making sure each step, like risk analysis and report writing, gets the attention

it needs. By investing in these early stages, we're setting up a strong foundation for the project's success. It shows that planning well is key to getting things done right and keeping costs in check. The cost per hour for one person has been chosen based on the different specialists required for the tasks. The most needed role in this stake is design analyst. UX designer is also very important in this process.

Implementation phase cost

The total cost for the Planning and Design phase is around £9,620.70. All of the three development milestones will have approximately the same number of hours spent on design, development, documentation, testing, and meetings. The sub-total cost for each milestone is expected to be around £3,206.90. The most used role for this phase is software developer. Since coding, documentation and coordination will highly depend on their decisions. The second most used role is QA engineer because a large amount of work in the development is spent on testing and ensuring the best quality of product is achieved.

User evaluation phase cost

The total cost for the user evaluation phase is around £5,816.90.

This phase includes designing the evaluation process, approving ethics, fulfilling participants' requirements, and analyzing the results.

Non-staff cost

The total amount of non-staff expenses is around £450.00.

This is also an important point to consider in terms of cost as these expenses often are not the focus. We expect to avoid spending on development equipment.

However, we expect to spend money on software subscriptions and pay participants at the user evaluation stage.

Total cost

The final cost of the presentation tool will be approximately £17,351.00.

We expect to spend 498 hours in total, which is around 100 hours per person.

10. Acceptance Tests

Feature	Test	Expected outcome

Undo	Clicking on ‘undo’ button.	Should undo the last action.
Redo	Clicking on ‘redo’ button.	Should redo the last action.
Save	Clicking on ‘save’ button.	Should the file in the selected directory.
Create new file	Clicking on ‘File’ menu and selecting ‘New file’	Creates new presentation file.
Open file	Clicking on ‘File’ menu and selecting ‘Open file’	Opens existing presentation file.
Presentation mode	Clicking on ‘present button’.	Should start the presentation mode in which we can view the slides in full screen and go to the next slide by pressing space.
Exit presentation mode	Pressing ESC on the keyboard.	Should exit presentation mode
Add slide	Clicking on ‘plus’ sign at the right side of the screen.	Should add a new empty slide just after the selected one.
Reorder slide	Holding on the slide in the thumbnail pane and dragging it in the needed direction.	Should give an opportunity to drag the slide and put it in the required order.
Find slide	Enter slide number into slide search box in thumbnail slide panel at the right.	Should go to the slide with entered input index.
Change file name	Clicking on the file name in the top right corner of screen.	Should give an option to modify the name of the file.
Toolbar sections	Clicking on the toolbar elements.	For each toolbar element it will pop-up the additional menu with more features
Copy	Pressing on copy button in the ‘Home’ menu or using Ctrl + C.	Should copy the selected element(s)
Paste	Pressing on paste button in the ‘Home’ menu or using Ctrl + V.	Should paste the copied element(s) in the clicked location.
Cut	Pressing on cut button in the ‘Home’ menu or using Ctrl + X.	Should cut the selected element(s).
Font style change	Clicking on font box in ‘Home’ menu. And selecting font.	Should show the drop-down menu with the selection of font. When the font is selected, the style of text element(s) selected on slides should be changed.

Font size change	Clicking on font size box in the 'Home' menu. Writing down font size and pressing enter or selecting different element.	Should give an opportunity to write the desired font of selected text element/s on the slide.
Font color change	Selecting text element and clicking on color box in the 'Home' menu and selecting required color.	Should change the color of the text.
RGB palate	Selecting required color in the RGB palette.	Should change the color of the selected element.
Bold font	Clicking on 'B' button in the 'Home' menu.	Should change the font of selected text element(s) to bold.
Italic font	Clicking on 'I' button in the 'Home' menu.	Should change font to italic.
Underline text	Clicking 'U' button and selecting color of underline	Should underline text with selected colour.
Edit/view mode	Clicking on the mode button	Should show drop-down menu under the button listing options of editing/viewing. When an option is selected it should enable the selected mode on the current file.
Search	Clicking on the search box at the top of the window.	Should give an opportunity to write text and it should search for it. Should have up/down arrows to go through the findings.
Searching by regular expressions	Clicking on the R button in the right side of search box.	Should enable regex mode in the search.
Scrolling through the slides	Using scroll on the mouse.	Should go through the slides list on the right-hand side of the screen or in the middle.
Zoom	Moving the slider right or left in the bottom right corner of the screen.	Should zoom in or zoom out the current slide.
Insert shape	Clicking on the shape in 'Insert' menu tab and selecting the needed shape.	Should give an opportunity to draw a shape on the slide.
Changing shape color	Clicking on the shape and then on the required color in 'Insert' menu tab. Clicking on the shape and then on border settings.	Should paint the shape into required color. Should give an opportunity to change width and color of the borders.
Inserting image	Clicking on the image button under 'Insert' tab and selecting required image file.	Should insert selected png/jpg image in the current slide.

Inserting table	Clicking on the table button under 'Insert' tab and selecting required table size n*n	Should insert table with selected size in the current slide.
Inserting hyperlink	Clicking on the hyperlink button under 'Insert' tab and writing link.	Should insert hyperlink into the current slide that should be open in the browser by using Ctrl+left click.
Moving objects	Holding the object on the slide and moving it in the required direction.	Should give an opportunity to change objects' position.
Speaker Notes	Before entering Presentation mode, the user adds speaker notes by pressing on the small microphone icon in the bottom left corner.	Should open notes menu for the current slide and give an option to add speaker notes.
Typing spell checker	Pressing on 'T' icon in the tool bar.	Should underline possibly wrong spelled words.
Tutorial	Pressing on Help button in the toolbar.	Should display the tutorial to the user on how to use the application.
Write code in application	Pressing on '#' in the toolbar.	Should open submenu with vs code to be able to write and compile the code in the presentation.

This covers testing of the most essential features of our product.

We expect the program to successfully pass these tests with expected behavior at the end of development.

11. Task List

Tasks	Team members
• Color blind mode • Write code in application • Typing spell checker • Tutorial	• Rhys • Maksym • Ivan • Rhys
• Open file • Create file • Save file • Edit/View mode • Undo • Redo • Copy/Paste/Cut elements • Shortcuts	• Maksym • Rhys • Ivan • Nik • Maksym • Maksym • Ivan • Ivan
• Add slide • Moving objects	• Maksym • Maksym

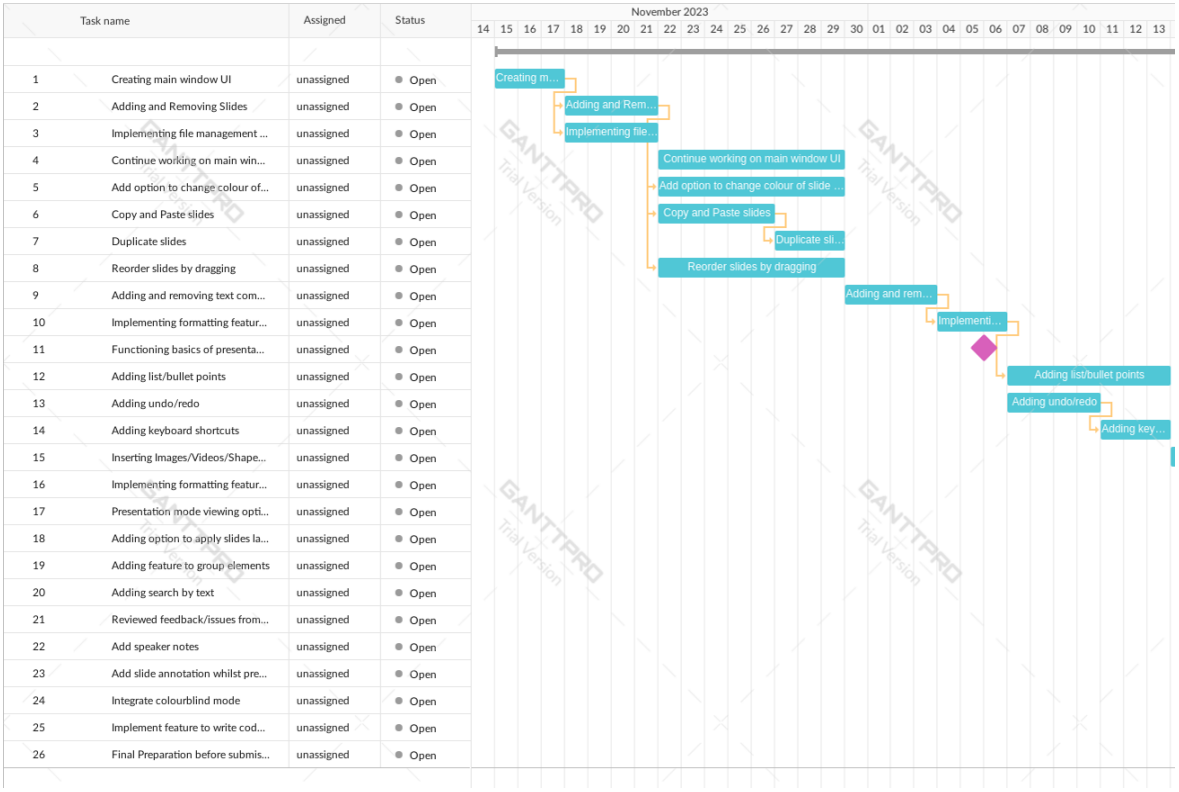
• Copy slide • Remove slide • Reorder slides • Zoom slides • Scroll though slides • Slide search by index • Search by text (+regex)	• Rhys • Ivan • Maksym • Maksym • Nik • Ivan • Maksym
• Insert tables • Insert images • Insert shapes • Insert charts • Insert hyperlinks	• Ivan • Rumaan • Ivan • Rhys • Maksym
• Change text font • Change text color • Change text size • Underline text • Bold text • Italic text • Text alignment • Text spacing • Text indentation • Bullet points • Numbered lists • RGB pallet to customize color	• Nik • Rhys • Ivan • Ivan • Ivan • Ivan • Nik • Rumaan • Rumaan • Ivan • Ivan • Maksym
• Present slides • Preview slides • Slide animations and transitions • Slide notations • Slide sound • Speaker notes • Slide templates and layout • Exit slide mode • Shortcuts in presentation mode • Presentation progress • Draw when presenting	• Maksym • Rumaan • Rumaan • Rhys • Rumaan • Rhys • Nik • Nik • Rhys • Maksym • Rumaan

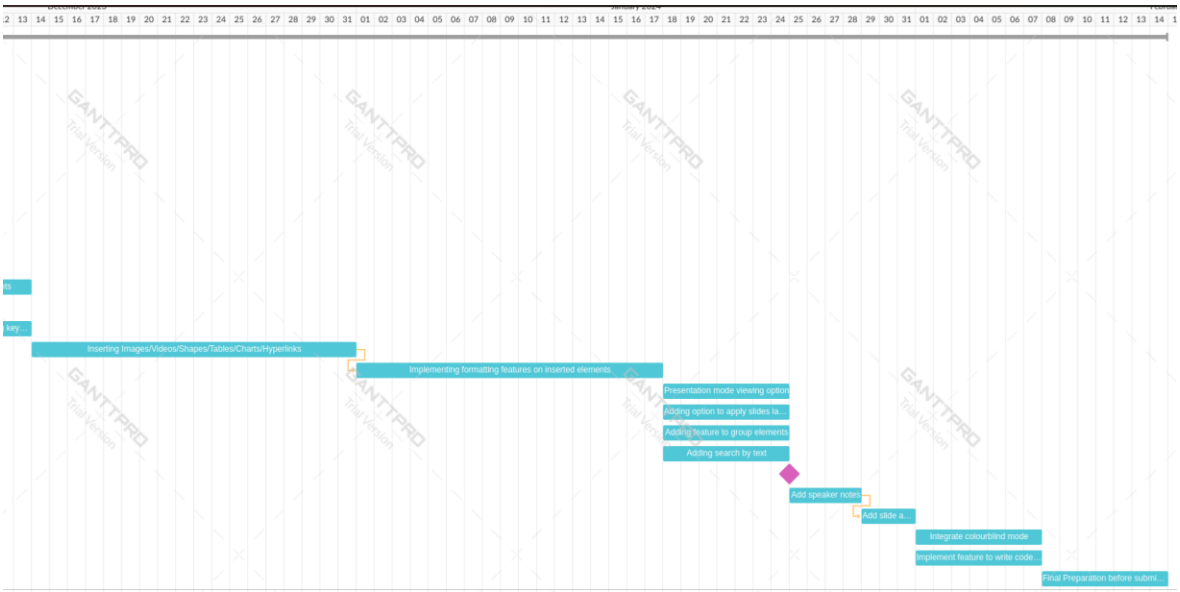
12. Activity Network

12.1 Gantt Chart

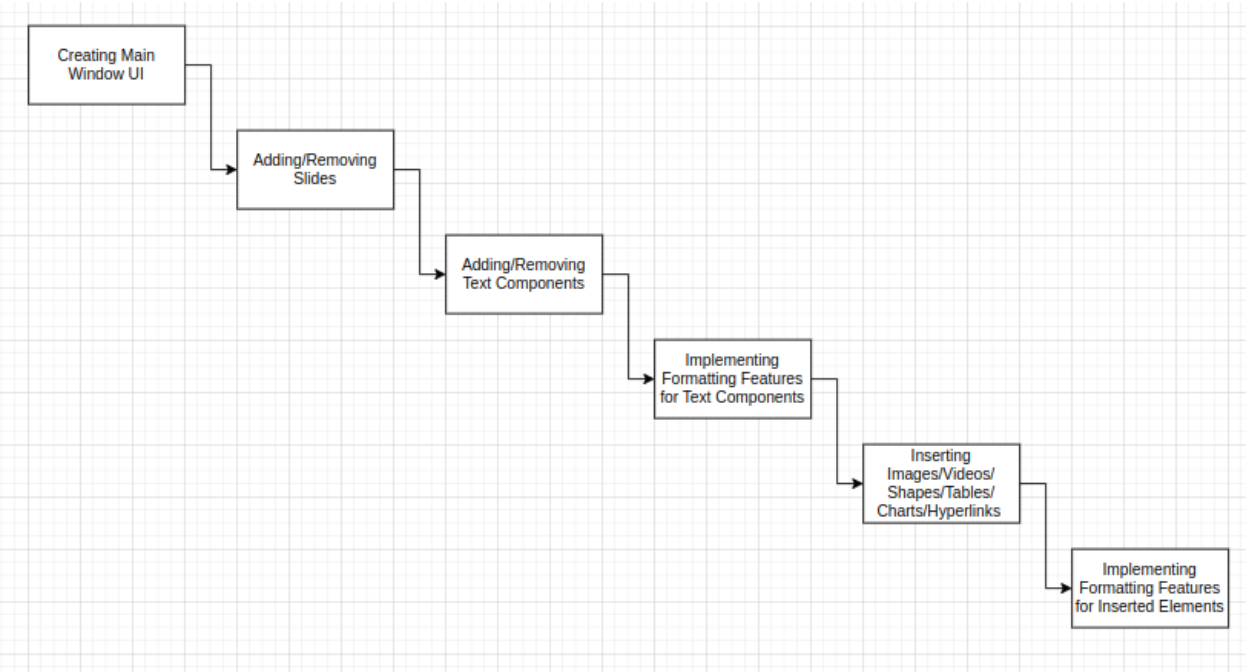
My Team | 210

	Task name	Assigned	Status
1	Creating main window UI	unassigned	Open
2	Adding and Removing Slides	unassigned	Open
3	Implementing file management ...	unassigned	Open
4	Continue working on main win...	unassigned	Open
5	Add option to change colour of...	unassigned	Open
6	Copy and Paste slides	unassigned	Open
7	Duplicate slides	unassigned	Open
8	Reorder slides by dragging	unassigned	Open
9	Adding and removing text com...	unassigned	Open
10	Implementing formatting featur...	unassigned	Open
11	Functioning basics of presenta...	unassigned	Open
12	Adding list/bullet points	unassigned	Open
13	Adding undo/redo	unassigned	Open
14	Adding keyboard shortcuts	unassigned	Open
15	Inserting Images/Videos/Shape...	unassigned	Open
16	Implementing formatting featur...	unassigned	Open
17	Presentation mode viewing opti...	unassigned	Open
18	Adding option to apply slides la...	unassigned	Open
19	Adding feature to group elements	unassigned	Open
20	Adding search by text	unassigned	Open
21	Reviewed feedback/issues from...	unassigned	Open
22	Add speaker notes	unassigned	Open
23	Add slide annotation whilst pre...	unassigned	Open
24	Integrate colourblind mode	unassigned	Open
25	Implement feature to write cod...	unassigned	Open
26	Final Preparation before submis...	unassigned	Open

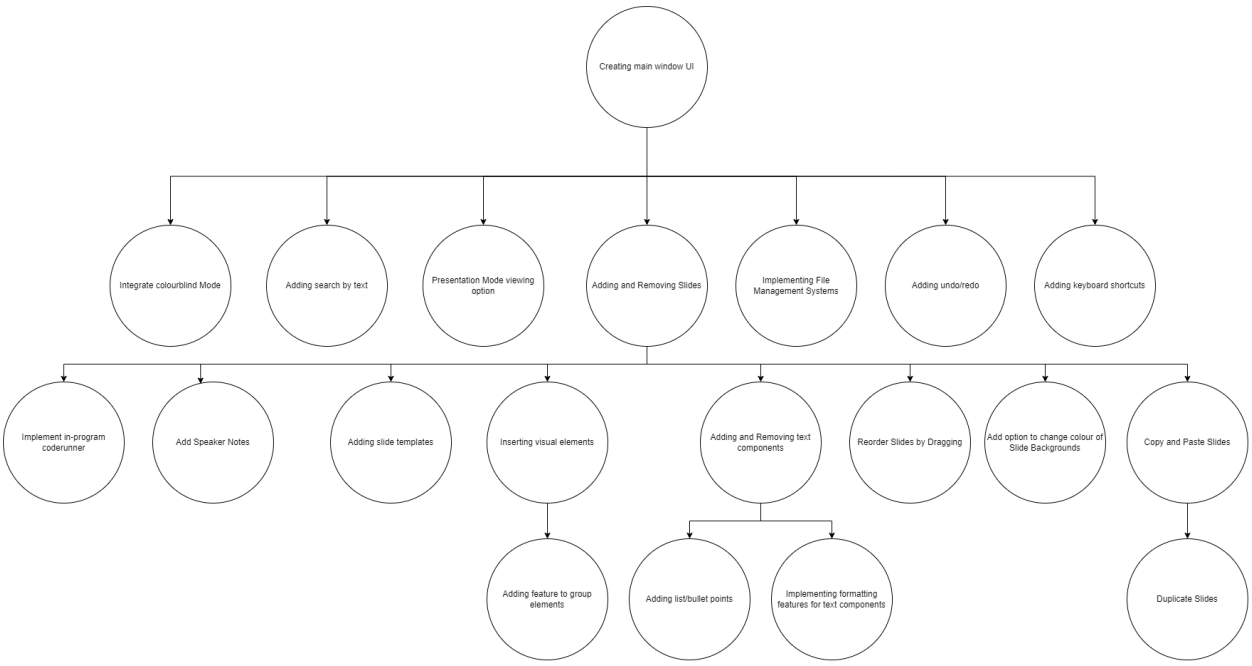




12.2 Critical Path Diagram

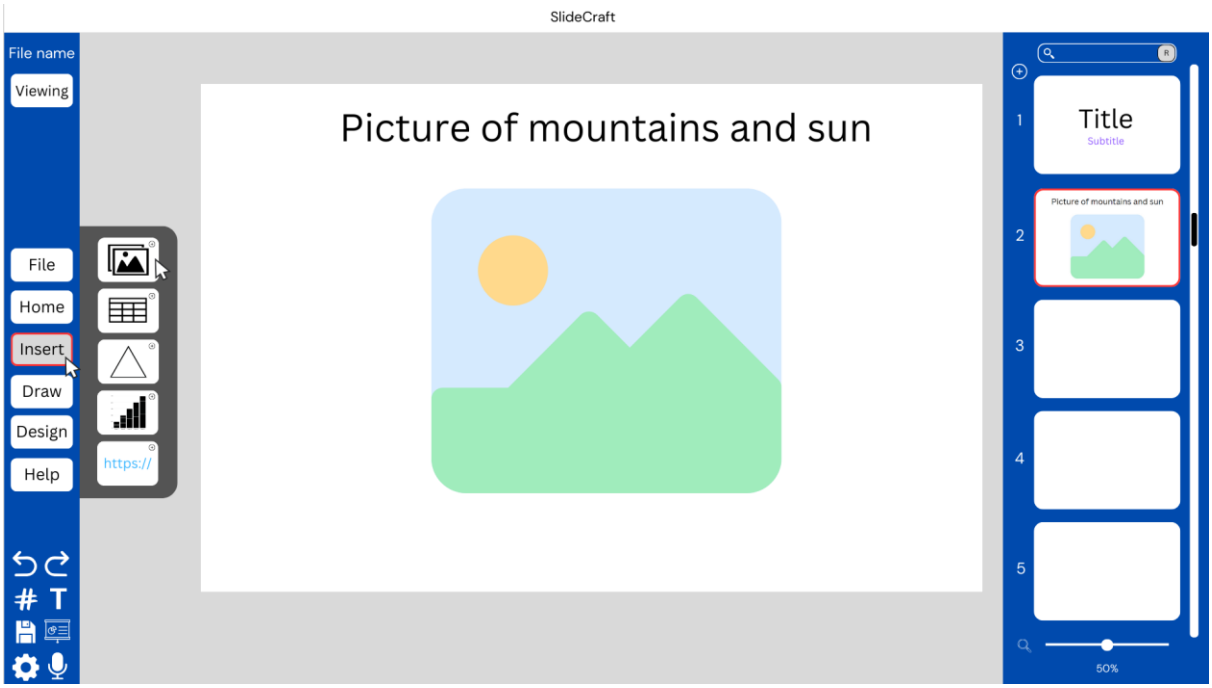


12.3 Activity Network Diagram



13. Appendix

13.1 UI


SlideCraft
File name
Viewing
File
Home
Insert
Draw
Design
Help
https://
Picture of mountains and sun
1
Title
Subtitle
2
Picture of mountains and sun
3
4
5
50%

Insert menu

13.2 UML Class Diagram