Early Deliverable for MSc Computer Science (Module Code: COMP8800)

Programme: MSC in Computer Science with an Industrial Placement

Topic: Mobile app for digital bereavement

Student name: Chun Yin Wong (cw692)

Supervisor: Jim Ang

**Overview of designs for griefSpace**

This document will demonstrate an overview of designs for the journal panel and insight panel using the Figma prototype. Although my early deliverable demonstrated a previous version of the designs, this document is needed to give the most updated prototype after some components are changed or dropped due to time constraints. If you are interested in a detailed Figma prototype, you can access it via the hyperlink below.

<https://www.figma.com/file/kNqOiNpYrLnY6EK2np6HS0/griefSpace?node-id=0%3A1>

1. **Overview of design**
   1. **Welcome page, introduction slider and its SwiperJS**
   2. Index page for the devices which is less than 768px wide

|  |  |  |
| --- | --- | --- |
| **Graphical user interface, text, application  Description automatically generated**  Figure 1.1.: First Slide with a welcome text | A picture containing icon  Description automatically generated  Figure 1.1.: Second Slide introducing Journalling system | Icon  Description automatically generated  Figure 1.1.3: Third Slide introducing visualisation  Graphical user interface, text, application, chat or text message  Description automatically generated  Figure 1.1.4: Forth Slide providing a CTA action to ask user to login |

Once the user opens the application on a mobile smartphone, the user will be directed to a page that will have four slides in total. The flow of user interaction is indicated by the four figures. First, user will see a welcome text and an ‘swiper’ left indicator in the first slide of the slider. Then, the second slide and third slide are used to introduce two main features to the user briefly. The final slide will have a Call-To-Action button which will end the journey by bringing the user to the login panel.

There are four indicators at the bottom of each slide telling user which slide they are currently in. Moreover, as this slider is handled by an external mobile-slider library Swiper JS, users can perform "swipe" actions to navigate through slide.

* 1. Index page for the devices which is at least than 768px wide

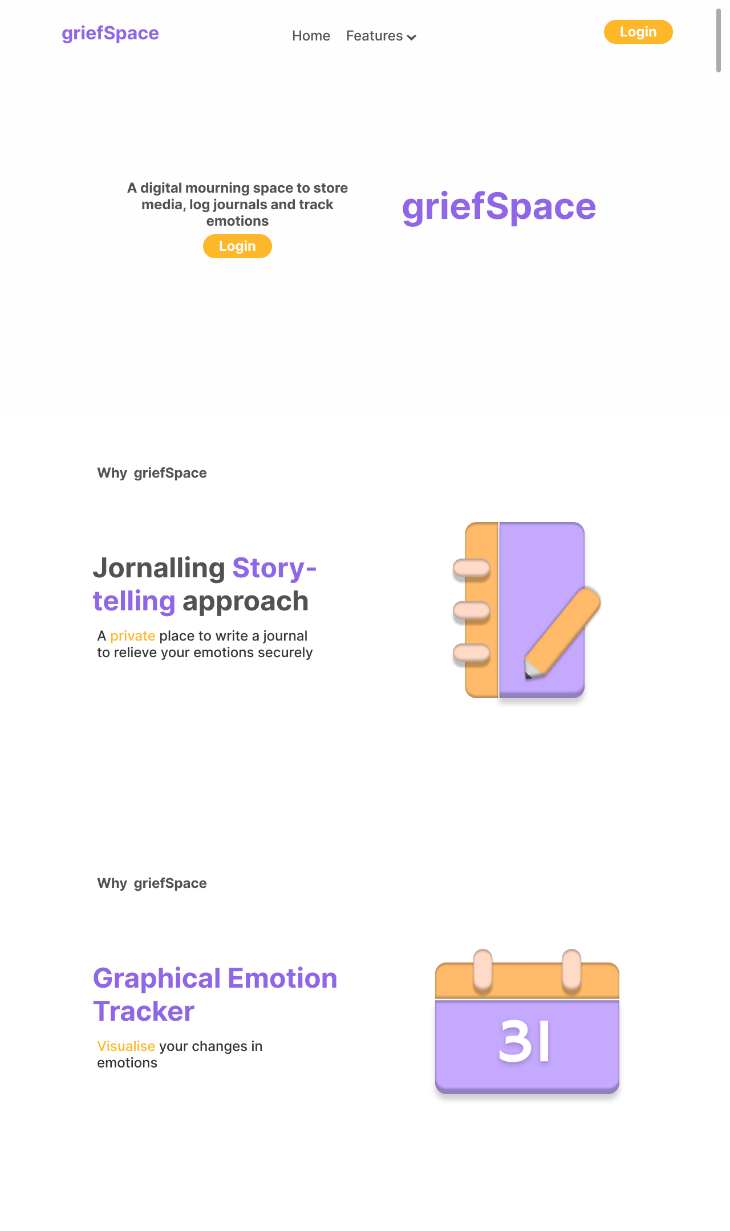


Figure 1.1.6: Home page for devices that is at least 768px wide

Figure 1.1.6 demonstrates the layout and design of a larger device. If the device’s width is at least 768px wide, the index page will make the bigHome container and the top navigation bar visible to the user. A media query is used to hide the mobile slider by setting its display attribute to display: none. This altering action aims to provide a better user interface design to the user. This bigHome container is designed to have three sections aiming first to ask users to log in, then introduce the main two features in two separate div.

* 1. **An overview of login and registration procedures**

1. Login page for smaller screen size

|  |  |
| --- | --- |
| Graphical user interface, text, application, chat or text message  Description automatically generated  Figure 1.2.1: Login page with Login Form | Graphical user interface, text, application  Description automatically generated  Figure 1.2.2: Login Page with Registration Form |

Figure 1.2.1 shows a login form on the login page. The login form will take two input fields, email address or username and password, to collect enough data to implement the identity confirmation. As the two-step authentication option is disabled by default, there will not be a consent screen requiring users to enter their OTP.

There will be a sentence at the bottom of the login form and registration form to allow users to perform a JavaScript action listener. At the bottom of Figure 1.2.1, the login form has a sentence stating, "Don't have an account? Register now", whilst Figure 1.2.2 shows a sentence stating "Already has an account? Login now" at the bottom of the registration form. Once the user clicks on these two sentences, the corresponding click listener will allow users to switch between the forms. For example, the login form shown in Figure 1.2.1 will leave the window, whereas the registration form in Figure 1.2.2 will be visible to users.

Login form aside, Figure 1.2.2 shows that the registration form will take four user inputs to finish the registration procedure. After the user clicks on the "register" button, values in all user inputs will be validated by a JavaScript script. If all the user inputs are valid, user data will be parsed into a relational database and inserted into a table. Finally, an HTML confirmation email with a one-time token and a hyperlink will be sent to the user email address to allow the authentication process. However, the login attempt will be rejected if invalid inputs such as missing values or invalid email format exist.

1. Login page for bigger screen size

Graphical user interface, application

Description automatically generatedGraphical user interface, text, application, website

Description automatically generated

Figure 1.2.3: Login form’s layout in bigger screen Figure 1.2.4: Registration form’s layout in bigger screen

Figure 1.2.3 and figure 1.2.4 demonstrate the layouts of the login form and registration form on a wider device respectively. As the login and registration forms are handled by the same corresponding Php scripts regardless of the display change, the bigger login page will only have different CSS styling. In addition, as shown in both figures, input fields in bigger screen sizes will have more width and larger margins to improve the readability of the layout.

* 1. **The Journal Panel**

A picture containing shape

Description automatically generated Treemap chart

Description automatically generated

Figure 1.3.1: Journal panel on smaller screen Figure 1.3.2: Journal panel on bigger screen

Figure 1.3.1 and figure 1.3.2 demonstrate the layouts of the journal panel on different devices. As shown in Figure 1.3.1, the journal panel will have three main sections: heading, journal boxes and bottom navigation bar. Once users enter this page, journal boxes will be dynamically rendered from the database. An external JavaScript will handle this dynamic rendering action whilst its functions will create elements to populate the HTML DOM. On the other hand, the navigation is designed to place at the bottom of the window as the bottom area is suggested to be a natural thumb zone. (SmahingMagazine, 2016). Another source also indicates that mobile heatmap can help improve user experience as more smartphones are used in the current era (VWO, 2022).

For the bigger screen, the journal page has a similar layout without substantially altering the layout. The only significant change is the change in the journal boxes’ width. As the create box is used as a Call-to-action button, the journal box with wider width aims to allow the user to click on the button more accurately and easily.

* 1. **The Preview Panel**

A screenshot of a phone

Description automatically generated with medium confidence Graphical user interface

Description automatically generated with medium confidence

Figure 1.4.2: Preview panel on bigger screen

Figure 1.4.1: Preview panel on smaller screen

Figures 1.4.1 and 1.4.2 demonstrate the layouts of the preview panel on different devices. As shown in Figure 1.4.1, the preview panel will have three main sections: details, question boxes and the bottom navigation bar. Once users enter this page, question boxes will be dynamically rendered from the database. An external JavaScript will handle this dynamic rendering action whilst its functions will create elements to populate the HTML DOM. Although the same function is used to fetch questions and answers, the preview panel will only display questions to avoid the web page becoming exaggeratedly long and difficult to scroll. On the other hand, there is also a fixed floating button to allow users to click and get to the edit panel. Moreover, the title and emotion will also be displayed at the top of the page to provide users with more transparent details.

For the bigger screen, this preview page has a similar layout without substantially altering the layout. The only significant change is in the journal boxes’ width and colour.

* 1. **The Details Panel**

Graphical user interface, text, application

Description automatically generated Graphical user interface, text, application

Description automatically generated

Figure 1.5.1: Details panel on smaller screen Figure 1.5.2: Details panel on smaller screen

Figures 1.5.1 and 1.5.2 demonstrate the layouts of the details panel on different devices. As shown in Figure 1.5.1, the details panel will have three main sections: details, answers and the bottom navigation bar. Once users enter this page, answers will be fetched from the database according to a given journal’s id and question’s id. The fetched answers will be dynamically appended to an answer container. Labels such as ‘Answer 1’ and ‘Answer 2’ are used to provide more precise information and reduce misunderstanding. As each question type can be answered multiple times, there can be more than one answer fetched. As a result, an additional sentence, ‘You’ve submitted 2 answers for this question,’ will indicate to users the number of entries fetched. Similar to the preview panel, a fixed floating button allows users to click and get to the edit panel.

For the bigger screen, this preview page has a similar layout without substantially altering the layout. The only significant change is removing the max-width attribute for the text.

* 1. **The Insight Panel and calendar**

Calendar

Description automatically generated with low confidence A picture containing graphical user interface

Description automatically generated

Figure 1.6.1: Insight panel on smaller screen Figure 1.6.2: Insight panel on bigger screen

Figures 1.6.1 and 1.6.2 demonstrate the layouts of the insight panel on different devices. As shown in Figure 1.6.1, the insight panel will have two main sections: the calendar and the bottom navigation bar. Once users enter this page, this page will fetch all the journals written by this user and retrieve unique emotion's id and timestamps. Another script will then fetch the emotion's context based on the emotions fetched from a table and replace the string with the corresponding hex colour code. For example, if the user creates a journal on that day, its day cell will be set to have a background colour based on the emotion string.

Moreover, each day cell is clickable to allow users to get its pointed preview panel. On the other hand, a container is also used to display all the available emotions and their corresponding colours. This container is used as an indicator to allow users to understand each colour's meaning better.

**Conclusion**

To conclude, this document illustrates the design of the *griefSpace* software. Programming languages such as JavaScript, PHP and SQL will be used in both frontend and backend development to enable interface communicating with relational database. In addition, frameworks and libraries such as Chart.js, SwiperJS, PHPMailer will be used to support necessary features.

If more detailed demonstrations on systems and features are needed, please find the document named ‘Brief introduction on features and the logic’ under the doc folder.