Title: Mobile app for digital bereavement

Student: Chun Yin WONG (James)

Kent login Id: cw692

Programme: MSc Computer Science

Supervisor: Jim Ang

**Project Logbook**

This document demonstrates the activities and meetings notes throughout the development of a software called ***griefSpace***. Please note that each activity is described briefly in several sentence as it is aimed to be a record of daily development. For more detailed information on the software, please visit the wiki and readme.md on the software repository:

<https://github.com/wcy692/griefSpace>

30th June to 7th June

1. Working on designing website prototypes using Figma

**7th July** – **weekly meeting**

* A meeting taken place between 1 and 2 in the afternoon
* Briefly describe which features to be includes and their functionalities
  + Login Page:
    - Login page is used as an index page where users will be on when they first visit the page.
    - For the mobile version where its screen width is smaller than 768px, users will see a mobile slider which probably includes a login/reg form, a div to describe what’s media page, a div to describe what’s journalling system and a div to describe what’s insight page.
    - For the device that is wider than 768px, mobile slider will be hidden but the index page will have a different style instead.
  + Media Page:
    - Media page is only available if users has logged on the system. Otherwise, users will be redirected to the login page.
    - Media page will have two main containers: one showing all the medias that users has uploaded, and one allowing users to upload media.
  + Setting page:
    - Setting page is only available if users has logged on the system. Otherwise, users will be redirected to the login page.
    - The initial ideas are that settings page includes a lot of settings such as changing colours, managing access, notification settings and multi-steps authentication.

1. **Tasks to be done before next meeting:**

* Try to figure out a simple 3 pages login and registration system that allows users to login, register and be able to see the message.

1. **Advice from Jim:**
   * Features that I’m interested in could be too much to develop and might eventually lead to a time shortage as I only have approximately a month before submitting the software.
   * Advice is to focus on completing journalling system and insight first as these are two main intentions for this application.

**8th July**

1. Complete a simple login system which allows users to login and register.
2. Start working on error handling and messaging.

**9th July**

1. Integrated login/reg system with error handlings that will check
   1. Password pattern and its length
   2. If username or email exists
2. Start working on forgetting password system as users might forget password due to human error

**10th July**

1. Add a feature to the current login/reg system to allow users to reset their password
   1. Users can now click on the ‘forget password’ hyperlink to get to a form where users can request a resetting password email.
   2. Once user request this email, a token will be generated and inserted into a separate table in the database.
   3. Users will then need to check their email inbox to retrieve the link to the resetting password form.
   4. Once users submit a request to update their password, time interval, token, user inputs will be validated.
2. Start working on the activation system.

**12th July**

1. Successfully integrate PHPMailer with the application
2. Email sent by PHPMailer will all be in HMTL format now.
3. Complete activation system
   1. Users will now receive an email once they register.
   2. Users will need to check their email inbox to receive an activation email
   3. The activation email will include description and a hyperlink to the activation form.
   4. Once users submit a request to update their password, time interval, token, user inputs will be validated.

**14th July** – **weekly meeting**

* A meeting taken place between 1 and 2 in the afternoon
* Briefly explain the 3-page simple login/reg system

1. **Tasks to be done before next meeting:**

* Send the design of the ‘Journal’ page to Jim via email to confirm the design, features, and the way to present to the users
* Make the login/reg system more application ready by adding CSS styling as the login/reg system is workable but hasn’t been styled

**16th July**

1. Finish learning a JavaScript external library, SwiperJS. This external library will be used on the index page to render a mobile slider and handle touching and swiper interactions.

**17th July**

1. Start working on designing the ‘Journal’ page on Figma. This Journal page will have several features
   * A container showcasing all the existing journals written by this user.
   * A modal page that allows users to create a journal
   * A page that allows users to preview a specific journal by fetching its journal’s id.
   * A modal page that allows users to update a specific journal

**19th July**

1. Send the Journal page’s design to Jim
2. Jim replied that the design looks good but needs to think about the functionality of tagging emotion.

**20th July**

1. Integrated the Swiper JS with the application. The mobile version of the index page now has a touch-friendly mobile slider.
2. Integrated the activation system and resetting password system into the application.
3. The application is now only developed based on 414px width. Responsive design will be done after finishing developing other essential features as CSS styling is likely to be altered during the development cycle.

**21st July** – **weekly meeting**

* A meeting taken place between 1 and 2 in the afternoon
* Briefly explain the mobile slider, activation and resetting password system
* Mobile Slider
  + Mobile slider now includes a login/reg form.
  + Considering 414px wide is not enough to show both login form and reg form in the same window, login and reg form are now separated into two containers.
  + A button is used to allow users to choose between the login form and the registration form.
* Activation System
  + Activation system allows users to receive an activation email after they successfully register on this website.
  + Once the users submit register, a binary token will be generated and hashed. This hashed token and its timestamp will be inserted into the ‘activation’ table.
* Resetting password system
  + This system allows users to receive an email to receive a hyperlink to access the related form.
  + Once the users request an email, a binary token will be generated and hashed. This hashed token and its timestamp will be inserted into the ‘forgetpwd’ table.
* Security, token and expiry
  + The non-hashed binary token will be converted to a hexadecimal token and concatenated with the activation page’s location. This string will be included in the activation email. To make it more secure, the token and the hyperlink are set to be expired after 30 minutes.
  + Once the user submits their activation request, script will the stored hexadecimal token from the URL to compare with the hashed token. The script will also compare the submit time to check if the request is expired.
    - 1. **Tasks to be done before next meeting:** 
         * Start working on the Journal system
         * My plan is to work on the ‘Journal’ page and its creating journal modal page first.

**22nd July**

1. Finished the ‘Journal’ page in which scripts will fetch all the existing journals created by the username.
2. The username will be fetched and stored as a global PHP $\_SESSION variable after the user successfully login. A session is created and used to allow this variable to be globally accessed across PHP pages.

**23rd July**

1. Start working on the ‘Creating journal’ page. The first version of the page will include two pages: one to let user select his emotion whilst another one will allow user selecting question and answer.
2. Constructing question by referring multiple sources and Living Memory Home.
3. The page on which users can select their emotion is now finished

**24th July**

1. Start working on the feature that allow user to answer more than one question. It might require the application to create element by listening to a button event. Further investigation requires.
2. The idea of creating element is now forgone as the action of updating HTML Dom is found to update textarea element. This behaviour caused the loss of data in multiple experiments.

**25th July**

1. The idea of creating element is now replaced by using session storage.
2. When users click on the ‘add question’ button, there will not be a newly created element. But instead, the question string and the answer string will be stored into the session storage.

**27th July**

1. Finished the first version of the ‘creating journal’ page.
2. The ‘save and exit’ feature is added to the system. Users can now click on the ‘save and exit’ button to save this journal.

**28th July** – **weekly meeting**

* A meeting taken place between 1 and 2 in the afternoon
* Briefly showcase the existing version of journal page

1. **Question from Jim:** 
   * Can users go back to the previous question and edit the answer when they are creating journal?
   * ***Ans:*** At this current stage, no, they can’t do that as the system now stores the previous data into the session storage. But the page allowing the user to enter answer is still the same page.

However, the user can edit the question in the ‘editing journal’ feature which will be done by next week.

1. **Advice from Jim:**

* Journal page:
  + - There are two before/go back button at the same page. These two buttons might cause users’ confusion and affect user experience. Therefore, one of the buttons might need to be removed to reduce the occurrence of confusion.
    - The editing problem of the existing ‘creating journal’ is acceptable as users can edit their question after they create the journal and submit the ‘edit’ request.

1. **Question from me:**
   * You once mentioned the visualisation should be more than pie chart in the feedback of early deliverables. I would like to know if there is any advice for visualisation despite using pie chart.
   * ***Ans:*** Can be a calendar in which each day will indicate the emotions of the users.
   * There can also be a bar chart in which its x-axis shows the time i.e., date whereas its y-axis shows the total amount of emotions counted for a period.
   * Pie chart can still be used. But there is a reminder that pie chart can only be usable to show a summary whilst the insight panel should be more than a summary.
2. **Tasks to be done before next meeting**
   * Confirm the design of the ‘insight’ panel with Jim
   * Complete the ‘Journal’ system

**29th July**

1. Agree that including a calendar can be useful as it can showcase the emotions.
2. The behaviour of the calendar emphasises the sense of timing and it might help highlight changes in emotion during a period.
3. Calendar can be handled by an external JavaScript library, fullCalendar.io. This library will handle the rendering of a calendar and allows to populate the data fetched from the database.

**31st July**

1. Drafting the first version of how the insight panel communicate with the database and populate the calendar.
2. The application will fetch the details of each existing journal from the database once the user enters the insight panel. Each emotion string will be replaced by a hex colour code. The colour code will then be used to be the background colour of a day cell.
3. Start working on the ‘edit journal’ page. However, considering users should preview and edit each question on a page, I might need to reconsider the creating element functionality.

**2nd August**

1. Finish the ‘preview page’, ‘details’ page and ‘edit journal’ page.
2. After users click on a journal on the ‘journal’ page, this journal’s id will be fetched and parsed to the preview page’s URL as a URL parameter.
3. The preview page will first get the URL parameter and fetch all the questions in this journal.
4. After users click on a question, the journal’s id and this question’s id will be fetched and parsed to the details page’s URL as a URL parameter.
5. The details page will then get the parameter and fetch and render its relevant answer.

**3rd August**

1. Finish a design of the ‘Insight’ page on Figma.
2. Finish a workable ‘Insight’ panel user interface.
3. This page now renders the calendar after the user enters the ‘Insight’ page.
4. Some of its day cells will be populated based on the data such as strings of emotions and timestamps from the database.

**4th August** – weekly meeting

* A meeting taken place between 1 and 2 in the afternoon
* Briefly explain the completed journalling system and the existing functions of the ‘Insight’ page.

1. **Advice from Jim:**
   * Insight and its readability

* Consider the visualisation impacts of the calendar
* For example, can each day cell represent multiple emotions?
* Multiple entry can make day cell too condensed to showcase details, leading to a negative effect in visualisation.
* ***Ans from me:*** Each day cell might only be able to represent one emotion in each day cell as more data could lead to unclear data and bad user experience.

As users can now create multiple journals in the same day, the ‘creating journal’ feature can integrate a creation time limit to only allow users to create one journal per day.

* + Media page
    - As the submission time is getting close, the idea of including a ‘Media’ page might need to be reviewed. Jim said it is acceptable to drop the ‘media’ page when considering the existing number of features included in this application.

1. **Tasks to be done before next meeting**

* To add a time limit
* Complete the back-end functionalities to allow the rendering of calendar
* Start working on the media page

**7th August**

1. Finish the ‘Insight’ system.
2. The user interaction’s flow is as follows:
   * After users enter this ‘insight’ page, scripts will fetch string of emotions and timestamps from the database. The emotion was tagged and selected by the user when creating the journal whereas the timestamp represents the creation time of a journal in epoch format.
   * Strings of emotion will form an array whereas each of them will later be searched and replaced by a hex colour code.
   * Different epoch timestamps will form an array whereas each element will be converted to a date format (i.e., YYYY-MM-DD). This specific format is used as it matches the date format used by fullCalendar.io.
   * A day cell will be set to use a background colour if the date it represents matches the timestamp.
   * Users can click on each day cell to get redirected to the preview page showing a specific journal.

**8th August**

1. Added a time limit functionality to the ‘creating journal’ feature to only allow users to create one journal per day.
   * This feature utilises a PHP DateTime class to create an object based on a epoch timestamp.
   * The epoch timestamp generates based on the current browser time and date.
   * The DateTime object will then be modified to get the beginning and the end of the day for that timestamp.
   * If current timestamp falls within the range, users has already created a journal today. The request of creating journal will be rejected.

**9th August**

1. Revise the error handling feature of Journal.php and editPanel.php
2. If there is an error existed after users attempt to update a journal, Journal.php and editPanel.php now will have a popup window to indicate the error message. Text in the div is based on the parameter and its values.
3. Add a floating window in the edit panel to alert users if the user attempts to exit without saving.
4. Add media query to general.css, topNav.css and login.css to handle responsive views for login.php
5. Add media query to modal.css to handle responsive views for editPanel.php and modal.php
6. The media query now only support those devices which are 768px-wide, more media query needs to be added to handle those devices which are wider.

**11th August**

1. Weekly meeting cancelled due to supervisor’s request.
2. Decided to drop the media page after several days of consideration because its feature to support uploading photos, audios and videos could require more time to develop. For example, the drag-and-drop function should be considered if it needs to be mobile-friendly.
3. On the other hand, according to some sources on stack overflow, the HMTL 'file' input tags need excessive time to investigate their compilability issues with the touch-screen device. Moreover, the JavaScript image gallery, instant preview on upload and compressing media also took more time than I expected.

**12th August**

* + - 1. Start working on the setting page.
      2. Created a UI where users can click to enter multiple sections such as accounts, push notifications and login settings.
      3. Users can now click on the ‘Account’ div to enter the account panel. Users will then be able to see their username and emails on this page. Email is changeable whilst username is unique and unchangeable.
      4. Two buttons, ‘change email’ and ‘change password’, allows users to execute two interactions.
      5. Users can navigate to the form by clicking the ‘change email’ button. The user will be required to enter a new email. Existing password is also required to confirm their identity.
      6. Users can navigate to a previously developed page where user can request a resetting password email by clicking the ‘change password’ button. Users will be required to enter their existing email to a ‘resetting password’ email with a one-time token in it.

**13th August**

1. Review the ‘creating journal’ system as it’s once considered users should be able to go back to the previous question and change their answer.
2. Rework ‘the creating journal’ system by implementing the concept used in the ‘editing journal’ system. The implementation of using session storage to store answer is now dropped. But instead, every time user clicks on the ‘add question’ button, a new element will be created. The new element will then be populated by a HTML template. The new element will then be appended to an existing HTML element. The function to append is used rather than to replace as it could help prevent the behaviour of refreshing DOM elements. It can accordingly avoid data inside textarea tags disappears.

**15th August**

1. Added a feature to the ‘setting’ panel to allow users to modify the setting of getting notifications.
2. The ‘push notification’ settings allow user to decide which events will trigger the function to send notification via email.
3. Integrated ‘changing email’, ‘creating journal’ and ‘updating journal’ systems with the ability to send email.
4. Added a feature to the ‘setting’ panel to allow users to modify the login-related activity.
5. Users can now switch on the login alert to receive an alert email in which login timestamp and the time zone will be specifies.
6. Users can now enable emailing one-time password to implement two-factor authentication.
7. Modify the login system by adding a form to allow users enter one-time password.
   * After user submit correct username and password, the application will send an email containing a 6-digit one-time password to the user’s email inbox.
   * The user will enter the received one-time password on otpChannel.php to verify their identity.

**16th August**

1. Add media query to the application to support devices up to 1366px.
2. login.php is now changed to index.php and moved to the root directory to allow future server to read the index.php automatically. The login/reg functions are now changed to handled by a different page. The system is tested and worked without error.
3. Modify the error handlings done by the setting as some of the conditional statement will redirect users to the same place and result in infinite loops. It is now changed to bring users to index.php

**17th August**

1. Writing logbook by converting the hand-written log to a digital word document.
2. Integrated meeting notes with the logbook.