DAILY PLANNER

FINAL PROJECT PRESENTATION

黃楷駿 Kai-Jun Huang (Kai)

張智華 Chih-Hua Zhang (Julius)

王瑋琮 Wei-Tsung Wang (Jason)

許育銘 Yu-Ming Xu (Min)

WORK DISTRIBUTION

Members	Work Distribution
張智華 Chih-Hua Zhang 黃楷駿 Kai-Jun Huang	UI, Style and Animation
	Data processing (Frontend)
	Comment (Frontend)
王瑋琮 Wei-Tsung Wang	SQL API (CRUD)
	Comment (SQLite)
	Translation
許育銘 Yu-Ming Xu	Comment (SQLite)
	Document and Slide

ISSUES

- An Useful App
- Time Management
- Easy to Memorize

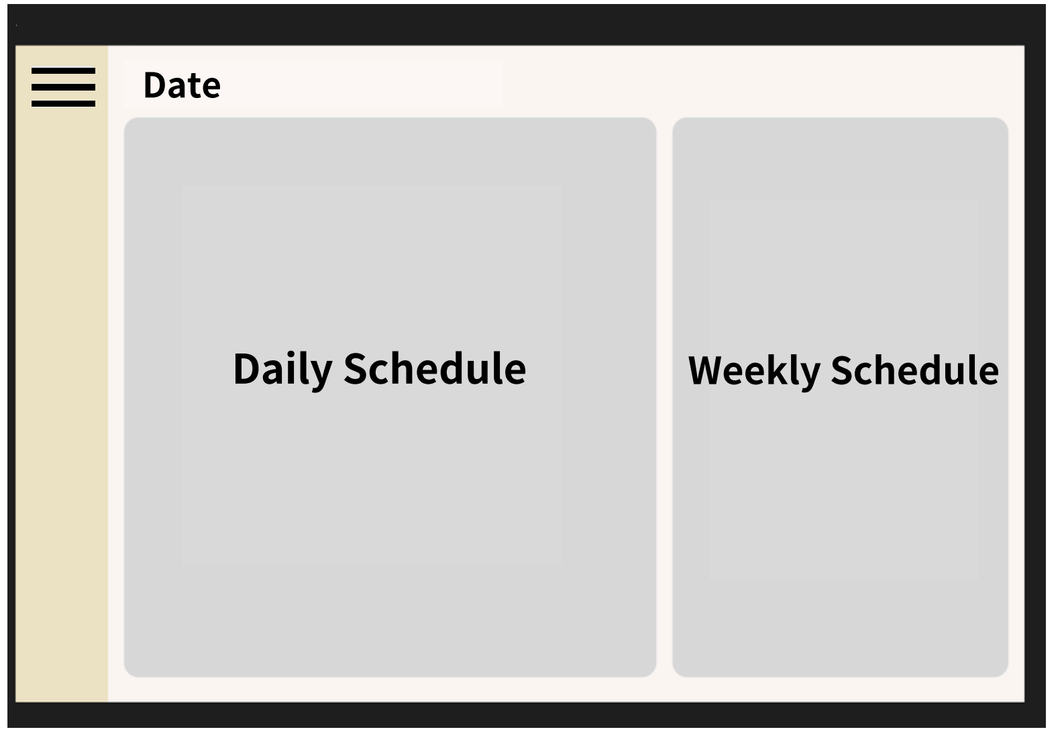
SURVEY

- Academic Tasks
- Daily Schedules
- Allocate Time

SOLUTION

OBJECTIVE

- Drawer
- Data Box
- Button



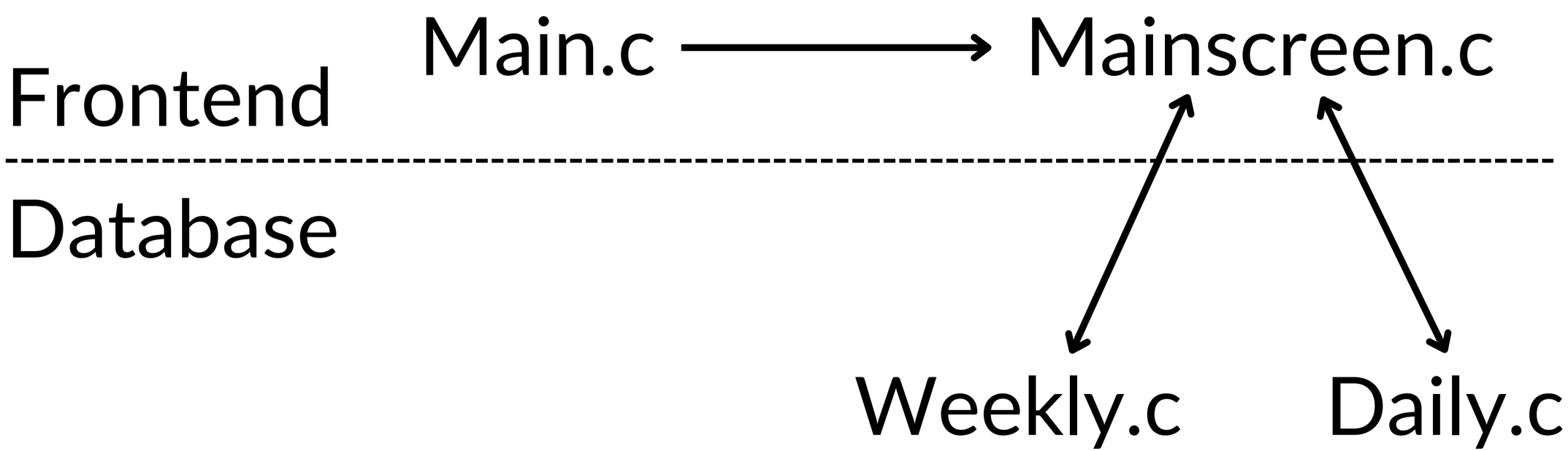
SOLUTION

DESIGN PHILOSOPHIES

- Allocate Time for Long-term Tasks.
- Work Efficiency and Quality of Life.

SOLUTION

SYSTEM ARCHITECTURE



IMPLEMENTATION ARCHITECTURE

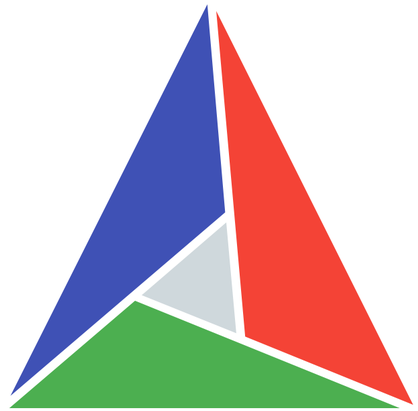
What is CMake?

- Cross-Platform build tool
- Generate platform-specific build flies

(Makeflies or Visual Studio project files)

What is CMakeLists.txt?

- CMake Configuration File
- Project Name, Source Files...



IMPLEMENTATION ARCHITECTURE

What does main.c do?

- Entry point of the program
- Define the core structure of application
- Load UI and CSS file

IMPLEMENTATION ARCHITECTURE

What does MainScreen.ui do?

- Defines components in Main Screen
- Drawer, Buttons, Menu...

What does MainScreen.css do?

- Styling Main Screen
- Border radius, Margins, Background color...

IMPLEMENTATION ARCHITECTURE

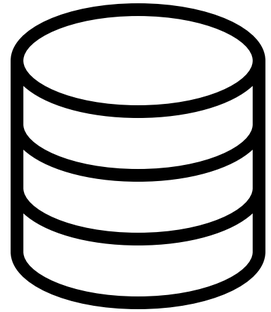
What does MainScreen.h do?

- Provides the interface MainScreen function in MainScreen.c to main.c

What does MainScreen.c do?

- Animations, Data Processing…

IMPLEMENTATION ARCHITECTURE



What does daily.h / weekly.h do?

- Provides the interface for SQLite APIs in daily.c / weekly.c to MainScreen.c

What does daily.c / weekly.c do?

- Provide database APIs
- Creating, Deleting, Retrieving and Updating

ACHIEVEMENTS

- Creating Application

```
38        GtkApplication *app = gtk_application_new("org.gtk.example", G_APPLICATION_DEFAULT_FLAGS); //創建app實例
```

- Creating Windows

```
6         GtkWindow *window = (GtkWindow *) gtk_application_window_new(app);
7         gtk_window_set_title(GTK_WINDOW(window), "Daily Planner");
8         gtk_widget_set_size_request(GTK_WIDGET(window),800,500);
```

- Loading UI file

```
12        GtkBuilder *mainBuilder = gtk_builder_new_from_file(ROOT_PATH"/src/MainScreen/MainScreen.ui");
```

- Loading CSS file

```
23        const char cssPath[]=ROOT_PATH"/src/MainScreen/MainScreen.css";
24        GtkCssProvider * cssProvider = gtk_css_provider_new();
25        gtk_css_provider_load_from_path(cssProvider, cssPath);
26        gtk_style_context_add_provider_for_display(gdk_display_get_default(),GTK_STYLE_PROVIDER(cssProvider),GTK_STYLE_PROVIDER_PRIORITY_USER);
```

ACHIEVEMENTS

- Drawer Animation

```
gboolean drawerAnimation(gpointer data)
{
    ...

    int dWidth = 20 * (isDrawerOpen ? -1 : 1);
    step++;

    int width = gtk_widget_get_width(drawer);
    width += dWidth;

    if (width > 330)
    {
        width = 330;
        step = 0;
    }
    if (width < 65)
    {
        width = 65;
        step = 0;
    }

    double drawerOpacity = (width - 65) * 1. / (330 - 65);
    gtk_widget_set_size_request(drawer, width, -1);
    gtk_widget_set_opacity(drawerContextScrolledWindow,
drawerOpacity);
}
```

- Show Drawer Context

```
if (!isDrawerContextShow)
    {
        drawerContextScrolledWindow = (GtkWidget *)gtk_scrolled_window_new();
        GtkWidget *drawerContext = (GtkWidget *)gtk_box_new((GtkOrientation)GTK_ORIENTATION_VERTICAL,
10);
        combine->drawer_context = drawerContext;
        gtk_scrolled_window_set_child((GtkScrolledWindow *)drawerContextScrolledWindow, drawerContext);
        gtk_scrolled_window_set_policy((GtkScrolledWindow *)drawerContextScrolledWindow,
GTK_POLICY_AUTOMATIC,
                                       GTK_POLICY_NEVER);
        gtk_scrolled_window_set_has_frame((GtkScrolledWindow *)drawerContextScrolledWindow, false);
        gtk_widget_set_name((GtkWidget *)drawerContextScrolledWindow, "DrawerContextScrolledWindow");
        gtk_widget_set_vexpand(drawerContextScrolledWindow, true);

        gtk_box_append((GtkBox *)drawer, drawerContextScrolledWindow);
        showDrawerContext(data);
        isDrawerContextShow = true;
    }
```

ACHIEVEMENTS

- Get selected week

```c
int calculate_iso_week_number(int year, int month, int
day)
    struct tm timeinfo = {0};
    timeinfo.tm_year = year - 1900;
    timeinfo.tm_mon = month - 1;
    timeinfo.tm_mday = day;
    timeinfo.tm_hour = 12;

    if (mktime(&timeinfo) == -1)
    {
        printf("Error: mktime failed to convert date.\n");
        return 0;
    }

    char buffer[3];
    strftime(buffer, sizeof(buffer), "%W", &timeinfo);
    return atoi(buffer);
}
```

- Connect Add Button Clicked Signal

```c
g_signal_connect(dailyAddButtonAnimationData->button, "clicked", G_CALLBACK(AddButtonOnClick),
                 dailyAddButtonAnimationData);
g_signal_connect(weeklyAddButtonAnimationData->button, "clicked",
G_CALLBACK(AddButtonOnClick), weeklyAddButtonAnimationData);
```

- Start Add Button Animation

```c
void AddButtonOnClick(GtkWidget *button, gpointer data)
{
    g_timeout_add(2, (GSourceFunc)bottomButtonAnimation,
data);
}
```

ACHIEVEMENTS

- Add Button Animation

```
gboolean bottomButtonAnimation(gpointer data)
{
  ...
  set_all_rounded_radius(tisData->backBox, dailyBoxRadius, tisData->type);
  gtk_widget_set_visible(tisData->Content_Scrolled, TRUE);
  gtk_widget_set_visible(tisData->contentBox, TRUE);
  gtk_widget_set_opacity((GtkWidget *)tisData->Content_Scrolled, (float)(expand - 50) / 350);
  gtk_widget_set_opacity((GtkWidget *)tisData->contentBox, (float)(expand - 50) / 350);
  gtk_widget_set_size_request(tisData->contentBox, -1, expand - 50);
  gtk_widget_set_size_request(tisData->Content_Scrolled, -1, expand);
  ...
}
```

- Add Activity to Box and Array

```
void v_daily_add(GtkWidget *button, gpointer data)
{
    ...

        v_D_create_widget(&D_data_regis);

        g_array_append_val(g_data, D_data_regis);

        Daily_Mission_Data_widget *regis = &g_array_index(g_data, Daily_Mission_Data_widget,
g_data->len - 1);

        gtk_box_append((GtkBox *)combine->daily_added_box, (GtkWidget *)regis->Display_Box);

        d_create_entry(&g_array_index(g_data, Daily_Mission_Data_widget, g_data->len - 1));

        v_signal_destroy(data);
        v_signal_connect(data);

        d_read_entries();
   ...
}
```

ACHIEVEMENTS

- Edit Activity

```
void v_W_revise_btn(GtkButton *button, gpointer data)
{
    ...
    g_timeout_add(2, (GSourceFunc)bottomButtonAnimation, combine->weekly_animation_data);
    v_W_update_widget(button, w_regis);
    gtk_button_set_label((GtkButton *)combine->weekly_widget->Add, "ADD");
    gtk_widget_set_sensitive((GtkWidget *)combine->weekly_widget->Activity, true);
    gtk_widget_set_sensitive((GtkWidget *)combine->weekly_animation_data->button, true);
    ...
}
```

- Remove Activity

```
void v_W_remove_widget(GtkButton *button, gpointer data)
{
    Mission_Combination *combine = data;
    GArray *g_data = combine->weekly_data;
    Weekly_Mission_Data_widget *w_regis;
    for (int i = 0; i < g_data->len; i++)
    {
        w_regis = &g_array_index(g_data, Weekly_Mission_Data_widget, i);
        if (w_regis->RemoveButton == button)
        {
            break;
        }
    }
    int removeIndex = 0;
    for (int i = 0; i < g_data->len; i++)
    {
        Weekly_Mission_Data_widget *d = &g_array_index(g_data, Weekly_Mission_Data_widget, i);
        if (d->Display_Box == w_regis->Display_Box)
        {
            gtk_box_remove((GtkBox *)combine->weekly_added_box, (GtkWidget *)d->Display_Box);
            removeIndex = i;
            break;
        }
    }
    g_array_remove_index(g_data, removeIndex);

    w_delete_entry(w_regis);
    w_read_entries();

    v_signal_destroy(data);
    v_signal_connect(data);
}
```

ACHIEVEMENTS

- Process +1 Button

```c
void v_D_update_widget(GtkButton *button, gpointer data)
{
    ...
    regis->now_frequency++;

    gtk_progress_bar_set_fraction((GtkProgressBar *)processBar, regis->now_frequency * 1. /
regis->frequency);

    int frequencyLength = 1, frequency = regis->frequency;
    while (frequency / 10 != 0)
    {
        frequencyLength++;
        frequency /= 10;
    }
    char frequencyText[frequencyLength * 2 + 4];
    sprintf(frequencyText, "%d / %d", regis->now_frequency, regis->frequency);

    gtk_label_set_label((GtkLabel *)frequencyLabel, frequencyText);

    if (regis->now_frequency >= regis->frequency)
    {
        gtk_widget_set_sensitive((GtkWidget *)regis->Add_frequency, false);
        gtk_button_set_label((GtkButton *)regis->Add_frequency, "Done");
    }

    d_update_entry(data);
    d_read_entries();
    ...
}
```

- Update Activity (SQL API)

```c
void d_update_entry(gpointer data)
{
    sqlite3 *db;
    char *err_msg = NULL;
    Daily_Mission_Data_widget *d_regis = data;
    char sql[strlen(d_regis->activity) + 200];

    if (sqlite3_open(DB_NAME, &db) != SQLITE_OK)
    {
        fprintf(stderr, "Cannot open database: %s\n", sqlite3_errmsg(db));
        return;
    }

    snprintf(sql, sizeof(sql),
             "UPDATE Calendar SET Now_Frequency = %d ,Frequency=%d, Begin_Hour=%d ,
Begin_Minute=%d , End_Hour=%d , End_Minute=%d WHERE Year = %d AND Month = %d AND Day = %d AND
Activity = '%s';",
             d_regis->now_frequency, d_regis->frequency, d_regis->begin_hour, d_regis-
>begin_minute, d_regis->end_hour, d_regis->end_minute, d_regis->year, d_regis->month, d_regis-
>day, d_regis->activity);

    if (sqlite3_exec(db, sql, 0, 0, &err_msg) != SQLITE_OK)
    {
        fprintf(stderr, "SQL error: %s\n", err_msg);
        sqlite3_free(err_msg);
    }
    else
    {
        printf("Entry updated successfully.\n\n");
    }

    sqlite3_close(db);
}
```

PROBLEM

Frontend:

1.Memory leak

2.Life cycle

3.Pointer

SQLite API (CRUD):

1.VS Code technical problem

2.Functions will be deprecated

BENCHMARKING

	Daily Planner	Google Calendar
To-Do list	O	O
Calendar	O	O
Sync Daily and Weekly Plans	O	X
Process visualization	O	X
Progress Check	O	X


THANK YOU
FOR LISTENING