

A thick grey vertical bar runs down the left side of the page. A light green arrow points to the right from the bar, containing the author's name. Below the bar, several thin, curved lines in shades of grey and blue sweep upwards from the bottom left corner.

Kseniia Skaletska

Final Project report

“Brain It On” App

Skaletska, Kseniia
300317947

REPORT NAVIGATION

App description.....2

Database Design.....2

Launching.....3

Main Page.....5

Task Description Page.....5

Edit Page.....6

Menu.....7

My Account.....7

Calendar.....8

Add Task.....9

List Of Tasks.....11

Conclusion.....11

BRAIN IT ON:

This project represents a to do application. The motivation of the app is to make doing tasks interactive for users by implementing achievements.

Among the features of this projects are:

- Database integration using SQLite
- CRUD on tasks and users
- Level Achievement
- Animations
- Notifications
- Calendar events

The App used such technologies as:

- **Android Studio** as IDE;
- **SQLite** for working with database
- **Java** for developing the functionality;
- **Nexus One API 24** as a virtual device
- **Photoshop** for designing the UI;
- **Illustrator** for drawing the icons and supplements;
- **Google** for inspiration and helping with arose questions

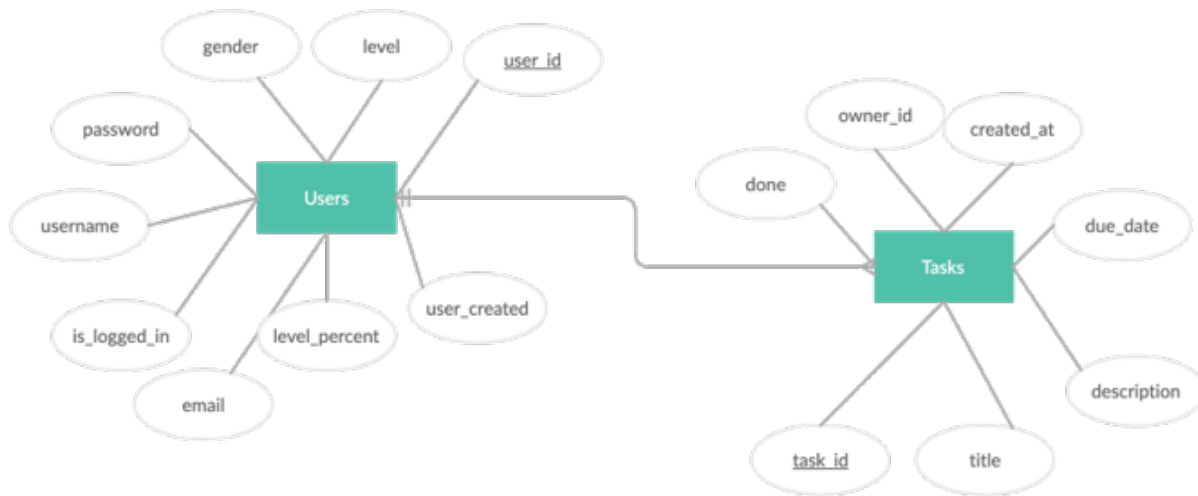
DATABASE DESIGN

Below you are able to see the ER diagram for the database used in the application.

The database includes two tables connected between each other. First table is for 'Users'. It has the relevant attributes and the Primary Key which is 'user_id'.

Another table is for 'Tasks'. It is connected with the 'Users' table via a foreign key 'owner_id', which is represented by the 'user_id' value in the 'Users' table.

The tables have the one-to-many relationship since one user can have many tasks, but the task can have only one user. Check the ERD below:



LAUNCHING

After opening the app, user will see the intro page with a 5 second animation of spinning brain. The unanimated design is shown in the Figure 1.

During Intro activity, the backend checks whether there is any logged in user. If any is found, the next activity will be the First Page (Figure 5), which is the Home page for the user. Or if none are found, the user will see the Login Page(Figure 3), where they can either log in or register(Figure 2) by clicking the “Sign Up button”.

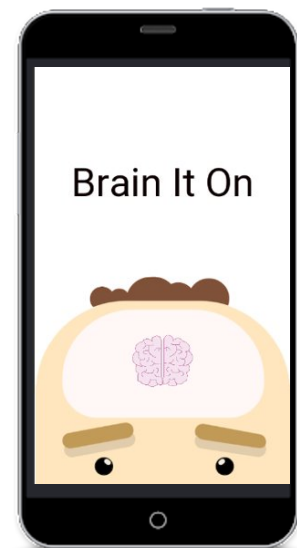


Figure 1. MainActivity class

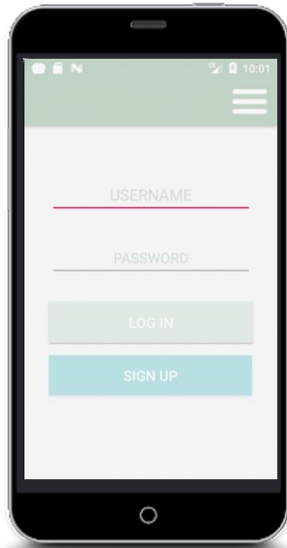


Figure 3 LoginPage class

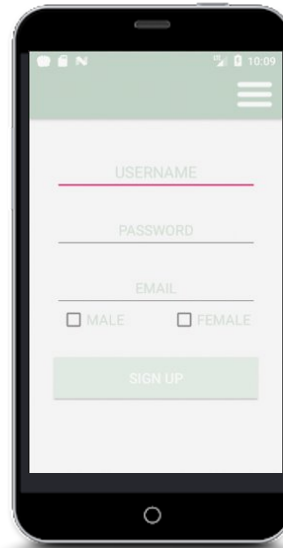


Figure 2 Register class

The register page creates new user. The needed information includes 'username', 'password', 'email' and 'gender'. Before putting new data into database, the password user entered will be hashed using MD5 to ensure the security of users.

When the user is already registered, the login page will check the entered username and password. The authentication process will first make the MD5 byte list of the password entered and then compare the value with the one in database related to this username. User will receive a notification on the phone (Figure 4) mentioning their username and will be transferred to the first page in case of the password being correct.

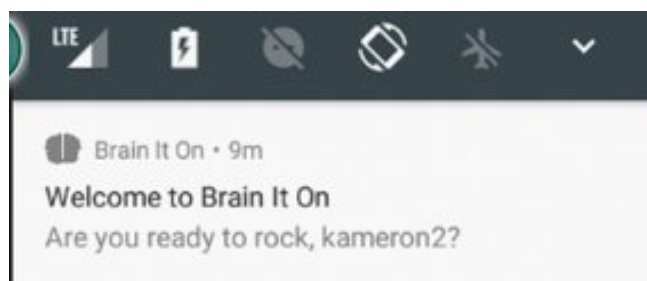
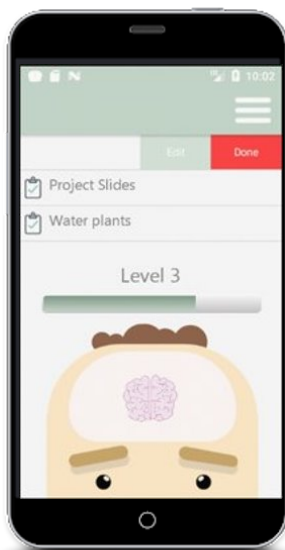


Figure 4 Notification after successful login

MAIN PAGE

The next thing this report will introduce is the FirstPage class(Figure 5) which shows the first 3 tasks which due soon, the level that users has, the progress bar with the level points and the picture with the brain, which will grow after every level achievement.



User can swipe the task left to see two options: Edit and Done. Whichever you choose will decide which activity is next. Also, user can click on the task to be transferred to the Task Description page(Figure 6).

TASK DESCRIPTION PAGE

On this page the user will be able to see all the details about the chosen task. At the very top is the title and the amount of days left till due date. In the code there is a calculation of “days left” made by deducting today’s date from the due date.

Since the description is optional it can be both present under the “days left” text widget or not present at all.



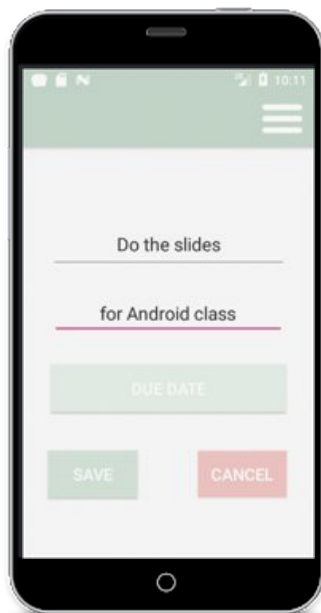
Also, this page contains a calendar with the due date highlighted, so the user can visually understand how long is left and what day of week it is.

Two buttons aligned at the bottom gives user two options on what to do with the task. It can be either done or edited. Button “Done” adds up points to the user level and marks this task done in database, so it will not appear in the due tasks anymore.

Figure 6 CreateUpdate class - edit task

EDIT PAGE

The “Edit” button will transfer user to the edit page(Figure 7). The Intent will include the id of the current task, so we can use it in the Create/Update class to find the task in database using dbManager and update it with the changed information.



Also, on the Edit page the task information will be added as a value to the relevant EditText, so the user can check the current data and amend it appropriately.

The “Due Date” button opens the calendar view for the user so he/she can choose the new date for the task or just check what the current due date is. The date is set to the due date of the task, so even if user don’t change its value, it will have same value as previous.

On this page, user has two options: either save the changes or cancel. Cancellation will change the Activity to the previous one and saving will update the task having the given in the intent task id.

MENU

On every page there is a burger-like menu button (Figure 8). By clicking on it, user will be able to see all pages that are in the app in a new activity.



Figure 8 Menu button

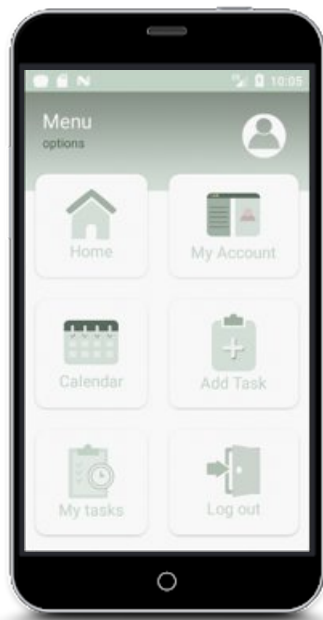


Figure 9 Menu class

The Menu Activity page (Figure 9) has the CardView API, which allows the app to have these nice rounded rectangles, which act like a button to transfer user to the relevant Activity.

There is total of 6 cardViews and all of them contain the icon drawn in Illustrator and the text explaining what Activity hides in this button.

The “Home” Button changes the Activity for the FirstPage class (Figure 5), which was introduced previously. All data will be up-to date.

MY ACCOUNT

The next button is “My Account”. This Activity displays the information about the logged in user. On Figure 10 illustrated the Account class.

At the very top, the user will be able to see the person’s icon. This is the icon which fits the user’s gender. Upon registration, the user will mark his/her gender and this information will be written to the database. The gender user chose decides what icon to show: male or female.

Under the icon, there is a level that user has and his/her username.

In the next section of this page there are two CardViews with the calculated number of tasks users need to do and the amount of done tasks.

Also, there is an email of the user present on the page along with two buttons which can transfer user to the Register page with the user info in the relevant fields or Log out.

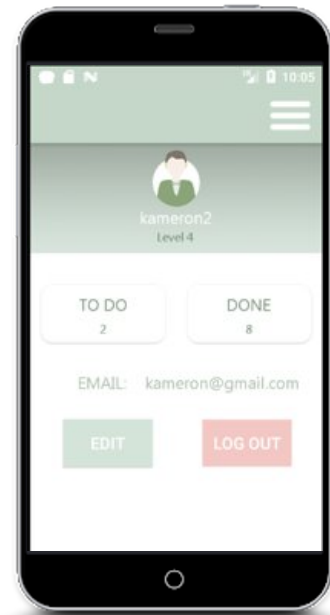


Figure 10 Account class

“Log Out” buttons change the value of the logged_in column of the user to 0, which stands for false. This is done in order to know that the user has logged out and before checking this info again he/she needs to login first. After the value of the logged_in columns is changed in database, the user will be transferred to the LoginPage, where the correct credentials needed.

CALENDAR

Back to the Menu page, user is able to see the Calendar button. By clicking on it, the Activity will be changed to the Calendar Class(Figure 11) where Calendar API is used.

In this activity users sees the calendar with the current date and the highlighted ones. The current date has a purple color and the days with the tasks are blue. For this Activity I user an API to highlight dates since in the regular CalendarView this option is absent.

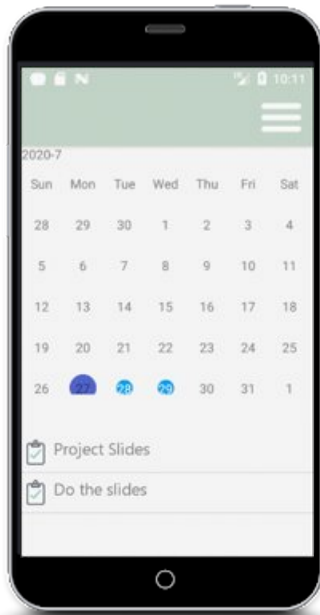


Figure 11 Calendar class

First, the app calls the `dbManager` method to find all the distinct dates that have due activities and then added those days to the calendar to highlight.

If person doesn't choose any date or chooses the date with no activities in there, the `ListView` below Calendar will be empty. But as soon as user clicks on the highlighted date, the app generates the list of the tasks which due that day and display it below the calendar.

The list of tasks is scrollable and also has a swipe option. Upon the swipe left action, the button "Edit" appears, so the user can switch to the edit activity of the chosen task.

The calendar itself also has a swipe option, so a user can navigate through different months and check whether anything is due next or previous month.

ADD TASK

Besides the presented buttons, there are several more in the Menu activity. Among them is Add Task activity (Figure 12).

On this activity user will see the empty `EditText` fields with hints what should be written in there.

The Title box is mandatory to be filled, since this is the main text of the task, which will be displayed in the listviews.

On Figure 13 displayed the calendarView which appears when user clicks on the Due Date button. On this view user will choose the due date for the activity by clicking on the relevant date in the calendar.

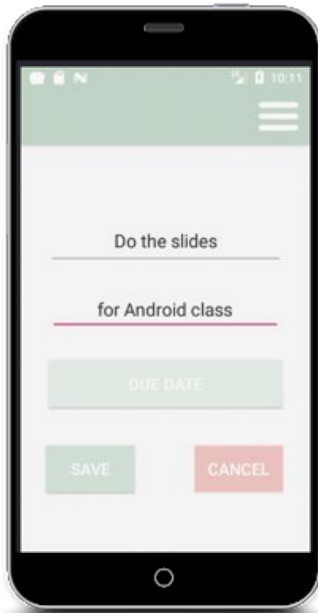


Figure 12 Add Task – with values

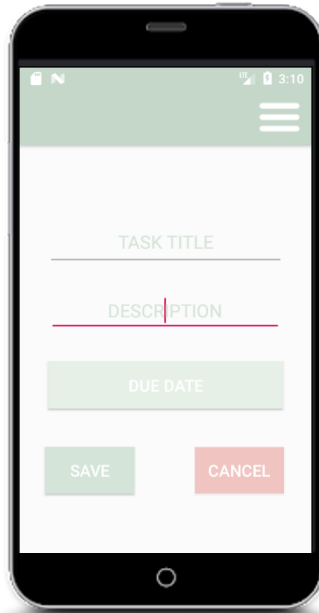


Figure 12 Add Task – without values

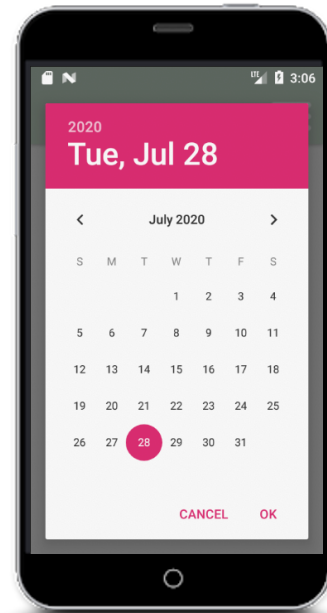


Figure 13 CalendarView

If the person tries to submit the form without entering the title, the Toast will appear on the page with the alert that this field is required (Figure 14)

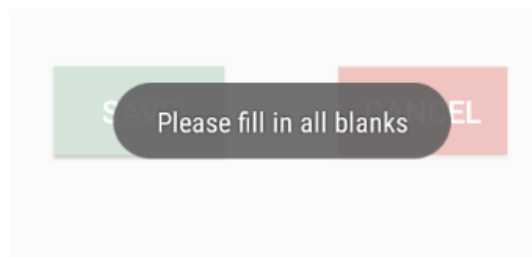


Figure 14 empty title toast

LIST OF TASKS

Another menu option is the Task List (Figure 16) where all the tasks are displayed in one list view. This View is scrollable and every item can be swiped left and show its options. However many tasks to do the logged user has, they will appear here.

In the app it is ensured that the tasks which are already done will not appear in any of the task list.

The database table for tasks has a column with the name “done” and the value of it is integer of 1 or 0. Digit ‘1’ is for ‘true’ and digit ‘0’ is surely for false. So while searching for user tasks, the dbManager sorts them, so the ‘done’ columns has value 0 and only.

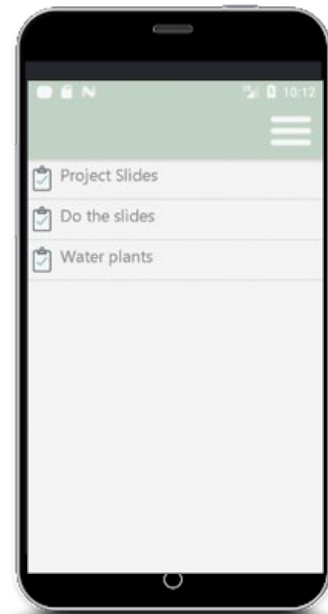


Figure 16 ListOfTasks class

CONCLUSION

While making this app I was trying to use as much covered in class material as possible as well as I made my best to implement the new interesting techniques which made my app more interactive and interesting for users.

Two of most challenging things for me in this app were calendar integration with the app and mixing Linear Layout containing ListView for tasks with the Constraint layout for other widgets. However, this experience helped me to understand the Android development concepts more deeply.

Among the new concepts learnt:

- API usage
- Mixed layout
- Shapes
- Relation Database integration
- Customized toolbar
- Working with progress bar
- Serializing the object to send between activities
- Swipe Menus
- Replacing of deprecated `getYear()`-like methods
- Notification creating
- Creating a dashboard
- Custom checkboxes

REFERENCES

<https://developer.android.com/training/appbar/setting-up>

<https://blog.jakelee.co.uk/reducing-the-size-of-textview-drawablestart-drawableends-using-styles/>

<https://www.youtube.com/watch?v=LOcD1evBcSA&t=627s>

<https://www.dignitasdigital.com/blog/how-to-make-your-progress-bar-as-aesthetically-pleasing-as-your-android-application/>

<https://github.com/baoyongzhang/SwipeMenuListView>

<https://www.youtube.com/watch?v=onyAgaZzYDg>

<https://stackoverflow.com/questions/2139134/how-to-send-an-object-from-one-android-activity-to-another-using-intents>

<https://teamtreehouse.com/community/solved-parcelable-encountered-ioexception-writing-serializable-object-name-packageweatherday>

<https://github.com/SpongeBobSun/mCalendarView>

<https://stackoverflow.com/questions/3972568/java-date-getdate-deprecated-refactored-to-use-calendar-but-looks-ugly>

<https://www.youtube.com/watch?v=ATERxKKORbY>