

Пояснительная записка к заочному этапу

конкурса проектов выпускников IT Академии Samsung

**InTime**

Участники: Черных Александер, Мазурова Варвара

Преподаватель: Степанов Павел Валерьевич

Город, площадка: Москва, РТУ МИРЭА

май, 2022 г.

**InTime** – приложение, которое помогает людям рационально распределять свое время. В современном быстроразвивающемся мире нужно идти в ногу со временем и рационально распределять время, ведь именно этот факт является залогом успешной жизни. Наше приложение предоставляет возможность пользователю фиксировать время своих занятий и отслеживать частоту того или иного занятия, это мотивирует людей двигаться к совей цели, поскольку мелкими шагами каждый день человек с большей вероятностью достигнет своих целей. В нашем приложении есть возможность отслеживать свои успехи в разделе «Календарь», где показано в какие дни были активности, чтобы по прошествию некоторого времени пользователь мог вспомнить, чем он занимался и какие у него были успехи, если он будет добавлять описание к своим активностям.

В разделе **«категории»** представлен список стандартных категорий, а именно «спорт» «музыка» «учеба» «работа» «хобби» «семья» «другое». Нам показалось, что это наиболее частые категории под которые попадает большинство наших повседневных занятий, если же ничего из перечисленного не подходит, то есть возможность добавить свое занятие в «другое». После перехода на одну из категорий пользователь может добавить занятие и далее активность, они все хранятся в базе данных, поэтому при добавлении они будут отображаться у пользователя в следующий раз, чтобы не добавлять нужное занятие каждый раз заново, а выбрать уже ранее добавленное. После добавление занятия пользователь переходит на раздел, где добавляет активность, где нужно выбрать время начала и конца, описание и название. Все эти данные попадают в базу данных, поэтому их можно будет посмотреть в разделе «календарь».

В разделе **«календарь»** пользователю представлен помесячный календарь, где отображаются его добавленные активности, у каждой записи есть информация о начале и конце, описание и название занятия, из которого берется данная активность. Дни, когда были действия помечены специальными обозначениями, чтобы было нагляднее. Есть возможность листать календарь и смотреть свои действия за другие месяцы.

В разделе **«профиль»** пользователь может увидеть свою почту, под которой был совершен вход, далее он может посмотреть сколько активностей было выполнено им за все время, ниже пользователю представлена статистика по категориям, то есть есть возможность отследить какой категории пользователь уделяет больше всего времени. На наш взгляд это очень удобная функция, ведь она помогает в планировании своего дальнейшего времени, с учетом этих данных.

На **главном экране** показано текущее время в часах, минутах и секундах, по истечению суток ползунок обновляется и начинается отсчет заново. По кругу пользователь может видеть закрашенные секторы, которые отображают его запланированные или текущие активности. При нажатии на кнопку «о занятии» можно увидеть оставшееся время для совершения действия. И повторным – вернуться к часам.

**Структура**

**Экраны**

**SplashActivity** – экран заставки, который виден пользователю при каждом входе в приложение.

**MainActivity** – активность приложения, она содержит контейнер на весь экран, куда подгружаются дальнейшие фрагменты.

**AuthenticationFragment** – фрагмент аутентификации, где пользователь может выбрать вход через гугл аккаунт или через почту, содержит текстовое поле – название приложение, картинку – логотип приложения и контейнер, где далее будут отображены поля для ввода почты и пароля.

**LoginFragment** – фрагмент, который содержит в себе 2 кнопки, а именно, кнопка для входа с гугл аккаунтом и кнопка для входа с почтой и паролем, этот фрагмент подгружается в контейнер, расположенный во фрагменте AuthenticationFragment.

**LoginEmailFragment** – фрагмент, который также подгружается на место контейнера во фрагменте AuthenticationFragment. В нем находятся 2 текстовых поля с подсказками, где пользователь должен ввести свою почту и пароль, далее есть 2 кнопки: вход и регистрация, если же пользователь ввел 4 раза неправильный пароль для существующей почты, то идет переход на фрагмент ForgotPasswordFragment.

**ForgotPasswordFragment** – фрагмент, на который идет переход при восстановлении и смены пароля, содержит текстовое поле с инструкцией, текстовое поля для введение почты, на которую будет отправлено письмо для смены пароля, при корректном заполнении требуется открыть письмо на почте, сменить пароль. Далее идет переход на фрагмент со входом через почту (LoginEmailFragment), где требуется ввести почту и новый пароль.

**CalendarFragment** – фрагмент, где пользователь может видеть календарь и существующие активности. Сам календарь – подключенная библиотека (<https://github.com/mahimrocky/EventCalender>), где есть возможность пролистывать месяцы и видеть в какие дни существовали активности пользователя. Ниже идет компонент ScrollView, где отображаются где активности за определенный день(при выборе дня в календаре), если активностей нет – это часть пуста, если же есть, то есть возможность пролистывания всех имеющихся, подгружаются активности с помощью дополнительного xml файла, который будет описан ниже.

**ActionsCalendar** – фрагмент, который помогает отображать активности на день через ScrollView во фрагменте с календарем, в него входят 4 текстовых поля, первое отвечает за название занятия, к которому относится активность, далее идет название самой активности, следом идет поле, которое отображает время начала и конца активности и последнее поля – описание активности, все эти данные берутся из базы данных.

**CategoriesFragment** – фрагмент, который виден пользователю при переходе на нижней панели на вкладку «категории». На фрагменте располагаются 7 статичных кнопки, которые нельзя изменить, то есть список категорий пользователь менять не может, при нажатии на одну из них идет переход на фрагмент CategoryFragment.

**CategoryFragment** – фрагмент, который показывает более узкую информацию о данной категории. Элементы отображаются с помощью ExpandableListView, родителями являются занятия, а дочерние элементы – активности. Вывод происходит через адаптер, который работает через HashMap, где ключи – строковые занятия, а значения представляют собой строковый список из активностей. Есть возможность сворачивать и разворачивать дочерние элементы. Внизу экрана есть кнопка с «+», которая дает возможность пользователю создать новое занятие с помощью диалогового окна, которое находится в фрагменте fragment\_item\_list\_dialog\_list\_dialog

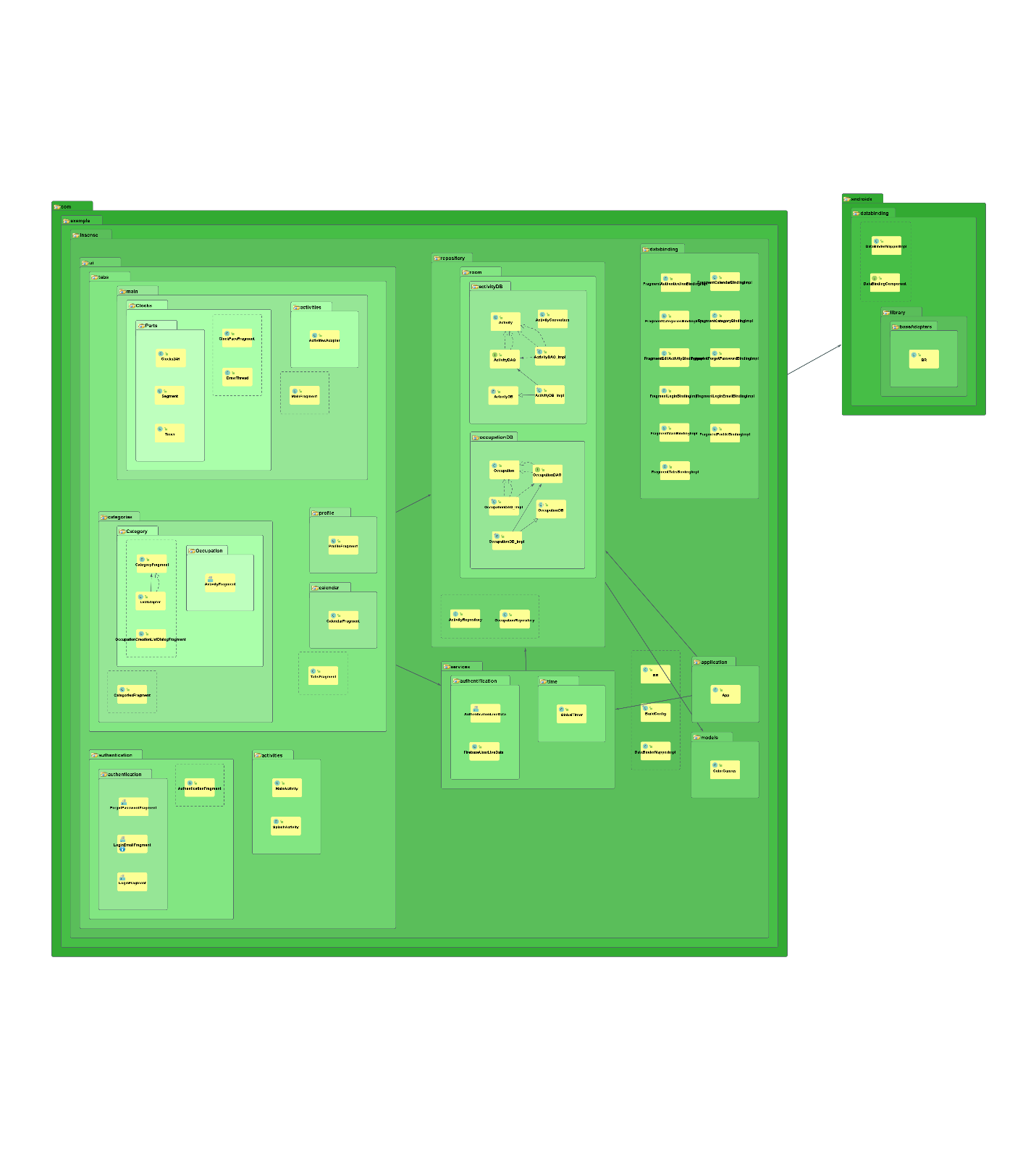
**fragment\_item\_list\_dialog\_list\_dialog** – фрагмент, где расположена разметка для добавления занятия. Есть 2 текстовых поля с подсказками «название» «описание», которые нужно заполнить и кнопка добавить для завершения процесса и добавления нового занятия.

**MainActivityFragment** – фрагмент, который позволяет создавать и редактировать свое занятие. Присутствует 3 текстовых поля наверху разметки, где нужно ввести название активности, занятие(выставляется автоматически, при создании новой активности и переходом на этот фрагмент) и описания. Ниже представлены две кнопки, первая отчеачает за вызов дилогового окна выбора дня и времени начала действия, а второе за время окончания действия. После ввода значений внизу есть 3 кнопки, которые позволяют вернуться назад, создать активность или удалить ее.

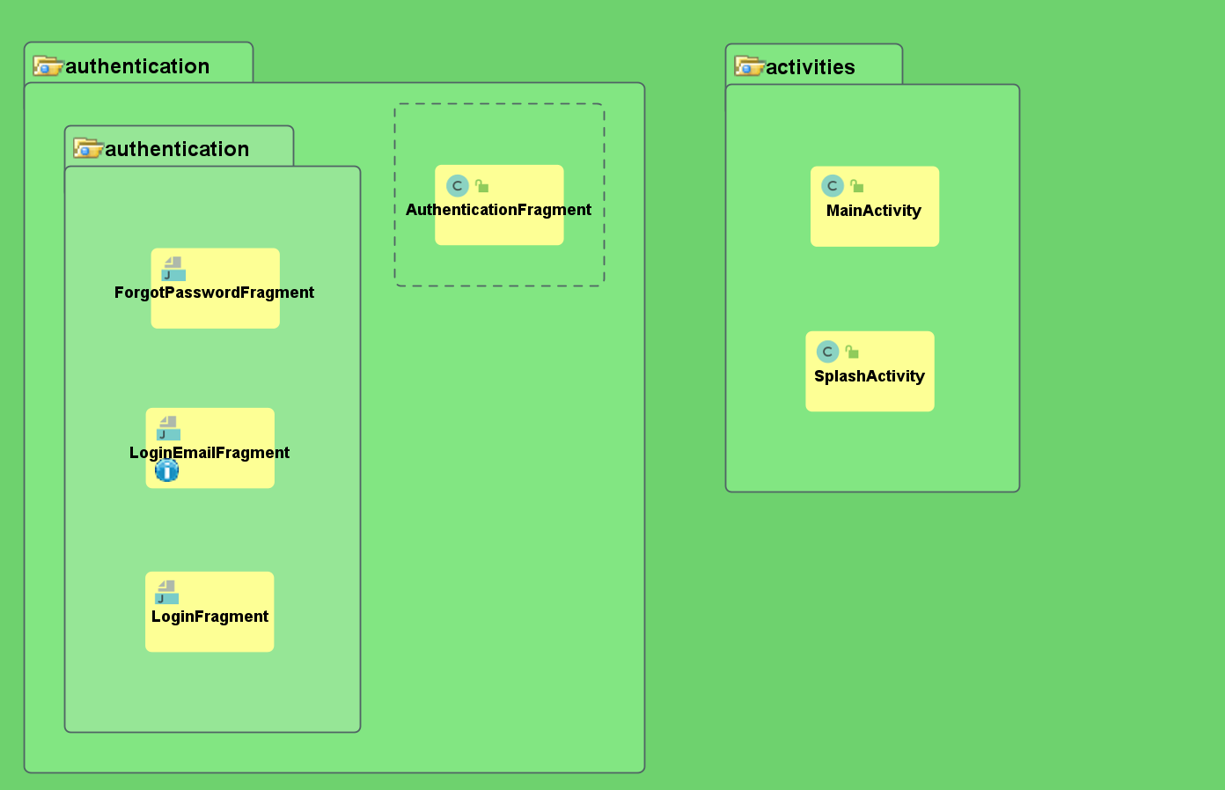
**ProfileFragment** – фрагмент, где видна информация о пользователе. Текстовое поле, где отображается почта, с помощью которой был совершен вход через firebase, текстовое поле, которое показывает все созданные активности пользователем, считаются с помощью базы данных. Далее есть статистика по категориям, каждому названию категории соответствует текстовое поле, где будет отображена цифра активностей, которые были созданы в этой категории, также работа с базой данных.

**MainFragment** – фрагмент, где можно посмотреть все дела на текущий день, и увидеть на часах примерное количество времени затрачиваемое на кадлое занятие, оно представлено в виде сегментов, угловой размер которых линейно зависит от затрачиваемого времени на каждое занятие, можно так же выбрать занятие, отредактировать его, и включить обратный отсчёт, который показывает – сколько времени осталось у пользователя на завершение занятия. Пришлось создать кастомное view, с использованием canvas, и использовать “сложную” математику.

Схема отношений между классами:

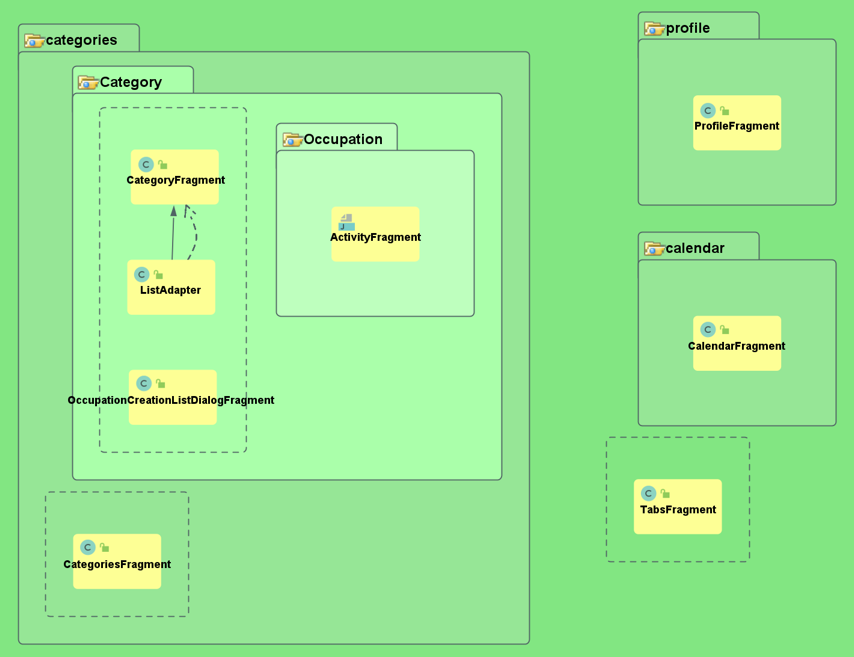


Более детальный разбор классов:



Слева представленны класса, отвечающие за регистрацию, ForgotPasswordFragment – класс, отвечающий за восстановление пароля пользователя, сверка логина и пароля идет через переменную типа FirebaseAuth. Класс LoginEmailFragment отвечает за вход и регистарцию пользователя с помощью пароля и почты. Класс LoginFragment – класс, отвечающий за предоставление выбора пользоватюлю аутентификации через гугл или почту.

MainActivity – главный класс программы, который содержит в себе контейнер для отображения других фрагментов, есть binding, который отслеживает нажатие на кнопку «о занятии». SplashActivity – активити, которая вызывается при старте приложение на 2 секунды, по сути класс пустой, содержит только показ фрагмента с иконкой приложения в течение некоторого времени.



CategoriesFragment – класс, который отвечает за определение категории, выбранной пользователем, используется binding и метод setOnClickListener для навешивания слушателей на кнопки, передача идет с помощью bundle и ключевого слова, которое передается на следующий фрагмент с помощью NavHostFragment.findNavController.

CategoryFragment – класс, отвечающий за вывод занятий и их дочерних элементов, то есть активностей с помощью ExpandableListView, для этого используется ListAdapter

**Заключение**

Командная работа над проектом велась с середины февраля за эти 4 месяца мы узнали много нового. Ни один из нас до этого не разрабатывал свои приложения и не сталкивался с такой средой разработки как Android Studio, но за эти месяцы каждый из нас понял, что это очень современная и удобная вещь. Наша работа начиналась с малого, первый класс, первый фрагмент, долго пытались разобраться в непонятных для нас методах, мы никак не могли подумать, что сможем сделать что-то подобное, с большим количеством класса, фрагментов и логики.

Мы и дальше будем улучшать наше приложение, например, хотели бы сделать дизайн чуть более красивым и привлекательным, также хотели бы добавить графики по неделям/месяцам/годам, чтобы пользователю можно было удобно отслеживать свои занятия. Также в связи с нарастающей популярностью Kotlin, мы думаем о том, чтобы впоследствии переписать наше приложение на этот язык. Хотим выразить благодарность нашему преподавателю за проделанную работу и помощь при возникновении вопросов и этому курсу за знания.

**Дополнительно**

Список сторонних библиотек/приложений, использованных для проекта:

* AndroidStudio (среда разработки для работы с платформой Android)
* Firebase (облачное хранилище, используется для входа с гугл аккаунтом)
* Room + SQLite (база данных)
* Библиотечный календарь событий (<https://github.com/mahimrocky/EventCalender>)
* Figma (дизайн)
* Компоненты из Material Design