# Министерство образования и науки Российской Федерации

# Федеральное государственное бюджетное образовательное учреждение

# высшего профессионального образования

# «Ижевский государственный технический университет имени М. Т. Калашникова»

# Кафедра «Программное обеспечение»

Отчет по лабораторной работе №2

по дисциплине «Конструирование ПО»

тема «Мобильный клиент для сервиса GitHub»

Выполнил

студент гр. Б08-191-2 С. А. Крюцин

Принял В.Г. Власов

Ижевск 2016

СОДЕРЖАНИЕ

1.ВВЕДЕНИЕ

1.1 Назначение системы

1.2 Область применения системы

1.3 Определения, акронимы, аббревиатуры

1.4 Ссылки

1.5 Обзор системы

2. ОБЩЕЕ ОПИСАНИЕ СИСТЕМЫ

2.1 Системный контекст

2.2 Режимы и состояния системы

2.3 Основные функциональные возможности системы

2.4 Основные условия системы

2.5 Основные ограничения системы

2.6 Характеристики пользователя

2.7 Допущения и зависимости

2.8 Оперативные сценарии

3. ФУНКЦИОНАЛЬНЫЕ ВОЗМОЖНОСТИ, УСЛОВИЯ И ОГРАНИЧЕНИЯ СИСТЕМЫ

3.1 Физические

3.1.1 Конструкция

3.1.2 Износостойкость

3.1.3 Адаптируемость

3.1.4 Условия окружающей среды

3.2 Рабочие характеристики системы

3.3 Безопасность системы

3.4 Информационный менеджмент

3.5 Работа системы

3.5.1 Эргономика системы

3.5.2 Ремонтопригодность системы

3.5.3 Надежность системы

3.6 Стратегия и регулирование

3.7 Устойчивость жизненного цикла системы

4. ИНТЕРФЕЙСЫ СИСТЕМЫ

5. КЛАССЫ КОДИРОВАНИЯ

5.1 UML схема.

5.2 Схема соответствий

Система: Мобильный клиент для работы с сервисом GitHub;

Подсистемы:

1. Внешний интерфейс.

2. Работа с базой данных.

3. Взаимодействие сервисом GitHub.

Модули:

1.1 Модуль вывода информации о репозитории.

Входные данные: Идентификатор репозитория.

Выходные данные: Форма вывода информации о репозитории.

1.2 Модуль вывода списка репозитория.

Входные данные: Идентификаторы репозиториев

Выходные данные: Форма вывода списка репозиториев.

1.3 Модуль авторизации пользователя.

Входные данные: Логин и пароль

Выходные данные: Информация об успешной или неуспешной авторизации

2.1 Добавление репозитория в базу.

Входные данные: информация о репозитории(владелец, наименования репозитория и т.д).

Выходные данные: сообщение об успешном или неуспешном выполнении операции.

2.2 Управление понравившимися репозиториями.

Входные данные: Идентификатор репозитория.

Выходные данные: сообщение об успешном или неуспешном выполнении операции.

2.3 Учет общего количества репозиториев владельца

Входные данные: Идентификатор владельца.

Выходные данные: Количество репозиториев в базе.

2.4 Поиск репозиториев среди имеющихся в базе.

Входные данные: Фраза, по которой необходимо найти репозиторий.

Выходные данные: репозиторий.

3.1 Аутентификация пользователей.

Входные данные: Логин, пароль.

Выходные данные: предоставление уровня доступа.

3.2 Обработка обновления списка репозиториев.

Входные данные: Количество подгружаемых репозиториев.

Выходные данные: список репозиториев.

3.3 Поиск новых репозиториев.

Входные данные: Фраза, по которой необходимо найти репозиторий.

Выходные данные: репозиторий.

3.4 Клонирование репозитория на девайс

Входные данные: идентификатор репозитория.

Выходные данные: файловое представление репозитория на девайсе.

3.5 Загрузка приватных репозиториев пользователя.

Входные данные: логин и пароль пользователя.

Выходные данные: список приватных репозиториев.

1. ВВЕДЕНИЕ

1.1 Назначение системы

Основное назначение системы является мобильный доступ к сервису GitHub с возможностью сохранить репозиторий на девайс. Мобильная версия только под Android.

1.2 Область применения системы

Система:Мобильный клиент для сервиса GitHub. Клиент должен предоставлять доступ пользователю к его аккаунту, включая приватный репозиторий. Так же необходимо реализовать список понравившихся репозиториев, которые можно выбрать среди общего списка случайно выбранных с сервиса GitHub списка репозиториев. Для удобства пользования должен быть имплемнтирован поиск по репозиториям. Если пользователь не имеет аккаунта в GitHub, то ему предоставляется возможность завести новый аккаунт, либо пользоваться неавторизированным пользователем, что исключает возможность видеть приватные репозитории, т.к. их нет.

Данная система будет применятся для решения следующих задач:

- мобильный доступ к сервису GitHub;

- возможность клонирования репозитория на девайсе;

- список репозиториев и поиск среди всех репозиториев.

Преимущества системы заключаются в следующих пунктах:

- Возможность пользоваться системой как авторизированным пользователем, так и неавторизированным.

- Возможность клонирования приватного репозитория на девайс.

-Material design и master/detail flow.

1.3 Определения, акронимы, аббревиатуры

Пользователь. Конечный пользователь системы.

1.4 Ссылки

1.5 Обзор системы

Система делиться на три главный модуля. Первый модуль отвечает за работу с сетью. Все запросы выполняются асинхронно и сохраняют результат в базу.

Второй модуль отвечает за работу с базой данных, необходим для хранения временных данных(выбор случайных репозиториев), а также для хранения данных связанных с пользователем(понравившиеся и приватные репозитории).

Третий модуль отвечает за вывод пользовательский интерфейс. Будет использоваться подход master/detail, где экран, находясь в горизонтальном положении, делится на два элемента, список элементов и конкретные детали по выбранному элементу.

2. ОБЩЕЕ ОПИСАНИЕ СИСТЕМЫ

2.1 Системный контекст

2.2 Режимы и состояния системы

Система может находиться в двух состояниях, которые соответствуют одному из состоянию в различных группах:

группа состояний 1:

Система может находиться в следующих состояниях:

- состояние системы в ожидании данных с сервиса gitHub;

- состояние системы в режиме просмотра списка репозиториев;

группа состоянии 2:

- состояние системы при авторизации пользователя;

- состояние системы при неавторизованном пользовании

2.3 Основные функциональные возможности системы



2.4 Основные условия системы

2.5 Основные ограничения системы

Максимальное количество скачиваемых за раз и выводимых репозиториев равняется 100 для гладкого и плавного отображения пользовательского интерфейса. Пользователю системы будет предоставлен выбора дизайна(цвета) и количество выводимых репозиториев.

2.6 Характеристики пользователя

Пользователь. Пользователь системы. Количество пользователей этого типа не ограничено. Использует систему для отображения понравившихся репозиториев.

Авторизированный пользователь. Пользователь системы. Количество пользователей этого типа не ограничено. Использует систему для отображения понравившихся репозиториев, а также приватных репозиториев пользователя.

2.7 Допущения и зависимости

2.8 Оперативные сценарии

Данная система подразумевает под собой следующий сценарий: пользователь авторизируется на сервисе gitHub, используя логин и пароль. Выбирает репозитории, которые ему понравились из списка случайно скачанных репозиториев. Клонирует репозитории, которые ему необходимы на девайсе, включая репозитории, которые доступны только этому пользователю на сервисе gitHub. Также имеется сценарий, где пользователь не авторизуется в системе. Тогда функционал будет похож, но не будет доступа к личным репозиториям авторизованного пользователя.

ФУНКЦИОНАЛЬНЫЕ ВОЗМОЖНОСТИ, УСЛОВИЯ И ОГРАНИЧЕНИЯ СИСТЕМЫ

3.1 Физические

3.1.1 Конструкция

3.1.2 Износостойкость

3.1.3 Адаптируемость

Основная нагрузка ложится на сервера GitHub.

3.1.4 Условия окружающей среды

Девайсы не зависят от погодных условий, если иного не указано в руководстве пользователя девайса.

3.2 Рабочие характеристики системы

3.3 Защита системы

При авторизации необходимо пройти проверку через систему reCapcha. Так же пользователь должен быть уверен в сохранении логина/пароля.

3.4 Информационный менеджмент

3.5 Работа системы

3.5.1 Эргономика системы

3.5.2 Эксплуатационная технологичность

3.5.3 Надежность системы

Сохранение логина/пароля будет проводится с помощью стандартного для Android AccountManager.

3.6 Стратегия и регулирование

3.7 Устойчивость жизненного цикла системы

Для устойчивости жизненного цикла системы используется обратная связь с пользователем.

4. СИСТЕМНЫЕ ИНТЕРФЕЙСЫ



Интерфейсы:

1. Интерфейс модуля работы с базой данных

**public interface** IDAO {  
 **void** insert(Repository repository);  
 **void** delete(**int** id);  
 **void** query(**int** id);  
  
}

2. Интерфейс модуля чтения данных из базы данных

**public interface** IReadDatabase {  
 **void** findRepository(String keyWord);  
 **void** getRepository(**int** id);  
  
}

44

3. Интерфейс модуля добавления данных в базу данных

**public interface** IWriteDataBase {  
 **void** saveRepository(Repository repository);  
 **void** cloneRepository(Repository repository);  
 **void** addToLikedRepository(Repository repository);  
}

4. Интерфейс модуля работы с сетью

**public interface** INetworkController {  
 IRegistration registration(User user);  
 IDownload downloadRepository(**int** idRepository);  
 IAuthorization authorization(User user);  
  
}

5. Интерфейс модуля авторизации в сервисе GitHub

**public interface** IAuthorization {  
 **void** authorizeUser(User user);  
}

6. Интерфейс модуля регистрации в сервисе GitHub

**public interface** IRegistration {  
 **void** registerUser(User user);  
}

7. Интерфейс загрузки репозитория

**public interface** IDownload {  
 List<Repository > downloadRepository(**int** idRepository);  
 List<Repository> searchRepositories(String keyWord);  
 }

8. Интерфейс пользовательского отображения

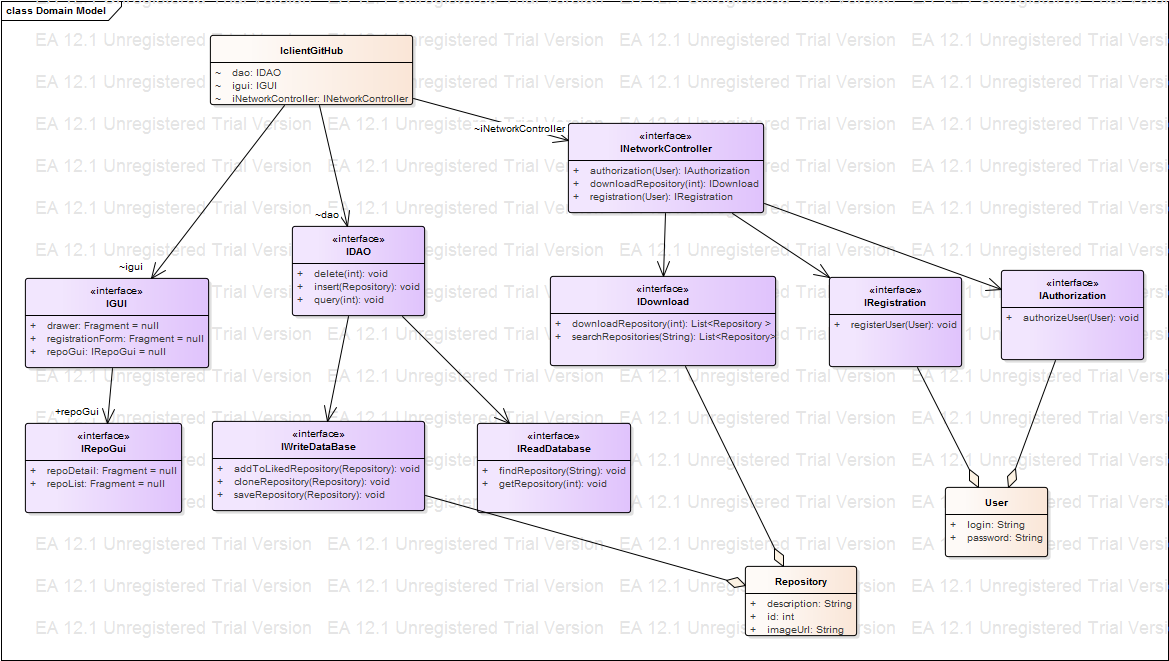
**public interface** IGUI {  
 Fragment ***drawer*** = **null**;  
 Fragment ***registrationForm*** = **null**;  
 IRepoGui ***repoGui*** = **null**;  
}

9.Интерфейс отображения репозитория.

**public interface** IRepoGui {  
 Fragment ***repoDetail*** = **null**;  
 Fragment ***repoList*** = **null**;  
}

5. КЛАССЫ КОДИРОВАНИЯ

5.1 UML схема программы



Классы кодирования

5.2 Классы кодирования программы соответствуют по функциональному назначению интерфейсам, по которым построены классы кодирования (см. п.4 Системные интерфейсы). Для системы так же разработаны следующие классы представления данных:

Класс репозиторий:

**public class** Repository {  
 **public** String **description**;  
 **public** String **imageUrl**;  
 **public int id**;  
}

Класс пользователь:

**public class** User {  
 **public** String **login**;  
 **public** String **password**;  
}

Тестирование.

Входные данные:

Неавторизованный пользователь, загрузка репозиториев с GitHub;