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

«Управление программными проектами»

Выполнили

ст. гр. Б08-191-2 Себельдин Д.В.

Принял Еланцев М.О.

Ижевск

2016

Подсистемы

1. Подсистема взаимодействия с приложением (API)  
   1.1 Модуль получения запросов  
   1.2 Модуль отправки ответов
2. Подсистема обработки данных.  
   1.2 Модуль обработки данных отправленных пользователем  
   1.4 Модуль генерации ответа пользователю
3. Подсистема push-уведомлений  
   2.1 Генерация события  
   2.2 Отправка на девайс пользователя
4. Подсистема хранения данных-

3.2 Модуль создания резервной копии базы данных

3.3 Модуль загрузки данных

3.4 Модуль сохранения данных

1. Подсистема идентификации  
   4.1 Модуль аутентификации  
   4.2 Модуль авторизации

Схема системы



Содержание

Перечень рисунков

Перечень таблиц

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

1.ВВЕДЕНИЕ

1.1 Назначение системы

Система предназначена для создания неформальных групп и проведения событий и мероприятий внутри группы.

1.2 Область применения системы

Серверная часть развертывается на операционных системах Windows c .net framework 4 и выше.

1.3 Обзор системы

Система представляет собой серверное приложение, работающее на компьютерах под управлением операционной системы Windows, на которой развернут net framework 4 и выше. Приложение обрабатывает запросы с мобильных устройств: создает группы, события, оповещает о приближающимся событии и т.д.

2. ОБЩЕЕ ОПИСАНИЕ СИСТЕМЫ

2.1 Системный контекст

2.2 Режимы и состояния системы

- получение запроса

- обработка данных

- сохранение или загрузка данных

- ответ от системы

2.3 Основные функциональные возможности системы

2.4 Основные условия системы

2.5 Основные ограничения системы

Перед работой с основным функционалом необходимо зарегистрироваться

2.6 Характеристики пользователя

Пользователь - конечный пользователь системы. Использует систему для создания событий, групп и мероприятий.

2.7 Допущения и зависимости

Оповещение о надвигающимся событии отправляется за сутки +/- 15 минут

2.8 Оперативные сценарии

Система предполагает следующий сценарий: пользователь создает событие или мероприятие, после чего другие пользователи смогут к нему присоединиться и получить уведомление за сутки перед событием

2.9 Основные функциональные возможности системы



3.1.2 Износостойкость

3.1.3 Адаптируемость

При какой-либо ошибке обработки изображения система информирует пользователя, прерывает текущую обработку и продолжает работу в нормальном режиме.

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

Система является полностью устойчивой и надежной

3.6 Стратегия и регулирование

3.7 Устойчивость жизненного цикла системы

4. ИНТЕРФЕЙСЫ СИСТЕМЫ

public interface IEventController

{

Answer CreateEvent(EventModel model);

IEnumerable<Event> FindEvents(EventFilter filter);

Answer RegistrationOnEvent(int userId, int eventId);

IEnumerable<Event> GetUserEvents(int userId);

}

public interface IGroupController

{

Answer CreateGroup(GroupModel model);

IEnumerable<Group> FindGroups(GroupFilter filter);

IEnumerable<Group> GetUserGroups(int userId);

}

public interface IDataStorage<T>

{

IEnumerable<T> Load();

void Save(T obj);

void MakeBackup();

}

public interface IIdentity

{

bool AddNewUser(RegistrationModel model);

bool Authentication(string email, string password);

bool PasswordReset(string email);

}

public interface IEventService

{

Event CreateEvent(User user, Group @group, EventModel model);

IEnumerable<Event> GetEvents(EventFilter filter);

Event GetById(int id);

}

public interface IGroupService

{

Group CreateGroup(User user, GroupModel model);

IEnumerable<Group> GetGroups(GroupFilter filter);

Group GetById(int id);

}

public interface IParticipantService

{

bool AddInGroup(User user, Group group);

bool AddInEvent(User user, Event @event);

User GetAllUsersGroup(Group group);

User GetUsersInEvent(Event @event);

}

public interface IUserManager

{

bool NewUserRegistration(UserModel model);

User FindById(long id);

bool LogOn();

}