ТЕХНИЧЕСКОЕ ЗАДАНИЕ НА РАЗРАБОТКУ  
Just In Time

|  |  |
| --- | --- |
| **Версия документа: 1.0** |  |
| **Дата создания**: 05.10.2015 |  |
| **Подготовлен:** | *Закиев Ауфар Азатович*  *Федоров Дмитрий Эдуардович*  *Залялов Ильдар Рафикович*  *Еднаков Даниил Александрович* **группа: 11-305**  **E-mail:**  [**daemonedge@mail.ru**](mailto:daemonedge@mail.ru)  **zaufar@gmail.com**  [**dartchief@yandex.ru**](mailto:dartchief@yandex.ru)  **Zalyalov.Ild@gmail.com** |
|  |  |

# 1. Введение

Just in Time - это приложения для чекинга студентов на парах. Данное конкретное приложение предназначено исключительно для пользования рядовыми студентами и старостами групп, не для преподавателей и деканата. Что делать с данными, полученными с помощью данного приложения, решает староста либо лицо, которому поручено отмечать присутствие студентов на парах.

Это приложение относится к области ведения отчетности и автоматизации контроля за группами людей. ПО такого типа позволяет экономить время ответственных за контроль людей и помогает отчитывающимся в удобной форме обмениваться информацией с контролирующими. Также это автоматизирует процесс создания отчетов и позволяет четко выделить зоны ответственности, что предотвращает конфликты между сотрудниками.

## Назначение

Документ Project Proposal содержит общие требования, описания и предложения в соответствии с теми характеристиками и решениями, которые могут быть использованы. Он содержи оценку проектных рисков и предложения по управлению этими рисками, решения возможных проблем, оценку времени и ресурсов проекта. Данный документ является отправной точкой для начала процесса утверждения проекта.

## Предмет

Предметом данного документа является продукт “Just in Time”, а также его особенности и требования к нему.

## 1.3 Термины, определения и соглашения

### *1.3.1 Аббревиатуры*

JIT - Just In Time, название проекта

ВШ ИТИС (ИТИС)- Высшая школа Информационных технологий и информационных систем

БД - База данных

### *1.3.2 Термины*

Leader – тот, кто отвечает за отмечание студентов на занятиях(староста или просто назначенное ответственное лицо)

# 2. Общее описание функциональности

Just in Time - это приложение для чекинга студентов на парах. В определенное время(начало пары) оно спрашивает студента, на паре ли он. Студент отвечает, и, в зависимости от ответа, происходят следующие действия - Да: проверка геолокацией местоположения и формирование соответствующего ответа, Нет: формирование соответствующего ответа. Далее идет отправка этого ответа на сервер. Ответ предназначен для старост групп. Их клиент получает данные в таблицу, которую старосты могут редактировать во избежание проблем, некоторые из которых описаны в Scope limitation.

## 2.1 Требования к функциональности

### *2.1.1 Категории пользователей*

Студент - обычный пользователь, не имеющий права вносить правки в базу данных  
Староста - староста группы, имеющий право редактировать столбцы в рамках таблицы своей группы

### *2.1.2 Функциональность*

### 1. Авторизация по своему аккаунту в VK или FB.

### 2. Создание аккаунта пользователя.

### 3. Отмечание присутствия студента на занятии

### 4. Редактирование журнала посещений

### 5. Показ расписания занятий

### *2.1.3 Практичность*

Пользователь имеет доступ к приложению через свое мобильное устройство. Не требует постоянного подключения к интернету - данные для авторизации будут храниться на самом мобильном устройств.

### *2.1.4 Надежность*

Система транзакций - для гарантирования доставки сообщений от пользователей типа “Студент” пользователям типа “Староста”.

Разделение отправляемых данных на малые части - для снижения нагрузки на мобильные сети и повышения энергосбережения.

### *2.1.5 Производительность*

Основные расчеты и преобразования происходят на сервере, вплоть до составления и хранения расписания занятий. Такой подход в купе с простым и понятным дизайном интерфейса гарантирует быстроту работу клиентской части на любых устройствах. Что же касается доступа в интернет, он требуется единоразово в течении каждой пары, что не обременительно, т.к. в большинстве ВУЗов имеется интернет с раздачей через   
WI-FI, что позволит проходить малозатратные процедуры геолокации и передачи оповещений.

### *2.1.6 Безопасность*

Ниже представлены основные требования безопасности:

1. Авторизация через соц. сеть перекладывает проблему с защищенностью на эти самые соц. сети.

2. Сервер обладает мощностью, достаточной для обеспечения бесперебойной работы приложения.

3. Никто, кроме старосты не сможет получить доступ к изменению данных в БД из-за того, что на каждую группу приходится только один возможный администратор(номер группы указывается при вводе личных данных)

4. Во временный окна, когда работа приложения не требуется, легко исправить ошибки с оборудованием

### *2.1.7 Требования к данным*

Приложение выводит интерактивные окошки и не требует вводить каких-либо данных для обычного пользователя. Работа с текстовыми данными производится на этапе заполнения таблицы студентов старостой.

### *2.1.8 Требования к преобразованию данных*

Данные преобразуются на сервере, единоразово, после чего отправляются всем пользователям, что позволяет сэкономить время и ресурсы памяти.

### *2.1.9 Масштабируемость*

Т.к. основными функциями приложения являются отправка оповещений о занятии и принятие и обработка сервером ответов, масштабируемость не требуется. Производственных мощностей среднего сервера хватит на столь не затратные операции.

### *2.1.10 Удобство поддержки*

На данном этапе обновления не запланированы. В случае изменения данной политики обновления будут производиться в соответствии с требованиями распространителей программного обеспечения устройств, на которых будет установлена клиентская часть приложения.

### *2.1.11 Требования к пользовательской документации*

Пользовательская документация на данном этапе построения приложения не требуется, интерфейс интуитивно понятен. В случае усложнения приложения – добавления новых средств для общения между leader и рядовыми пользователями документация будет добавлена в обновлении вместе с изменениями.

### *2.1.12 Требования к лицензированию*

Лицензирование требоваться не будет. В случае массового использования JIT в различных ВУЗах средства для лицензирования будут добавлены с обновлением приложения, лицензирование будет проводиться под надзором руководства ВУЗа, использующего данное приложения в качестве основного инструмента для отслеживания посещений.

# 3. Функциональное описание решения

## 3.1 Выбор компонентов, которые необходимо приобрести

Необходим обычный компьютер с установленной на нем ОС Linux Ubuntu (версия для серверов) либо сервер с той же ОС. Так же необходима лицензия разработчика под iOS и Windows Phone.

## 3.2 Интерфейсы

### *3.2.1 Пользовательские интерфейс*

В качестве основного примера рассмотрим интерфейс для приложения под iOS. Интерфейс состоит из 5 экранов:

1. Регистрация/входа

Экран вход под учетной записью. Имеется кнопка «Регистрация», в случае, если пользователь еще не зарегистрирован. В данном случае, экран обновляется и появляется форма для регистрации.

2. Главный экран

На нем отображено расписание на сегодня и статус пар (активная – идет сейчас, не активная – уже закончилась или еще не началась).

3. Оповещение

Это экран, появляющийся при входе в приложения по клику на оповещение о том, что занятие началось. Он представляет собой текст оповещения и две кнопки, являющиеся вариантами ответа на вопрос текста оповещения. При нажатии на одну из них происходит операция проверки с помощью геолокации, данный экран обновляется и появляется новое оповещение, наполнения текстового поля которого зависит от результата проверки местоположения пользователя.

4. Информация о приложении

Здесь записаны контактные данные создателей приложения и номер версии приложения.

5. Расписание

На данном экране расположено актуальное расписание на неделю.

Навигация между экранами реализована методом свайпов.

Внизу каждого экрана(кроме 3) расположена кнопка опций, в которых можно выйти из приложения на экран 1.

Там же расположена кнопка обновления расписания.

***3.2.2 Аппаратный интерфейс***

Есть сервер, вэб интерфейс на любом устройстве и клиентские приложения на мобильных устройствах. Информация преимущественно хранится на сервере. На клиентских устройствах может храниться закэшированная информация логин/пароль для входа по технологии remember me, а так же тексты стандартных оповещений.

### *3.2.3 Программный интерфейс*

Есть сервер и взаимодействующие с ним клиентские интерфейсы двух типов: для обычных юзеров и для leader. Сервер принимает и обрабатывает данные, приходящие с клиентских приложений и реагирует на них своевременно.

### *3.2.4 Коммуникационный интерфейс*

С сервером клиентские приложения взаимодействуют через сеть Интернет.

## 3.3 Обработка ошибок

1. Отсутствие интернета в момент оповещения – обновление экрана оповещения, появления на нем текста типа «Нет соединения с интернетом» и кнопки «Обновить»

2. Неверные данные логин/пароль – очищение полей ввода и вывешивание сообщения об ошибке ввода данных непосредственно рядом с полями ввода.

3. Ошибка геолокации – постоянное обновление операции до успешной проверки или истечения времени ожидания, после чего вывешивание сообщения «Ошибка в работе геолокации» и кнопка «Обновить».

## 3.4 Планирование работ

1. Обновление расписания при входе позднее заданного время(какое-то время до начала первой пары) .

2. Проверка обновлений приложения в случае добавления данной возможности.

3. Обновление общего расписания в случае его изменения на сервере(перенос пар).

4. Геолокация – во время требования проверки местоположения

## 3.5 Источник данных

База данных одна, состоит из нескольких сущностей(более подробно рассмотрено в работе IDEF1X.

Получение данных о местоположении реализуется средствами геолокационных инструментов устройства, на котором установлена клиентская версия приложения.

## 3.6 Проектные требования и ограничения

Проект предполагает использование языка программирования Ruby и его фрэймворка Ruby on Rails для написания кода сервера приложения и вэб-интерфейса для Leader’a, Objective-C для написания клиентского приложения под iOS и С# с фрэймворком MVVM для написание клиентского приложения под Windows Phone.

Ограничения:

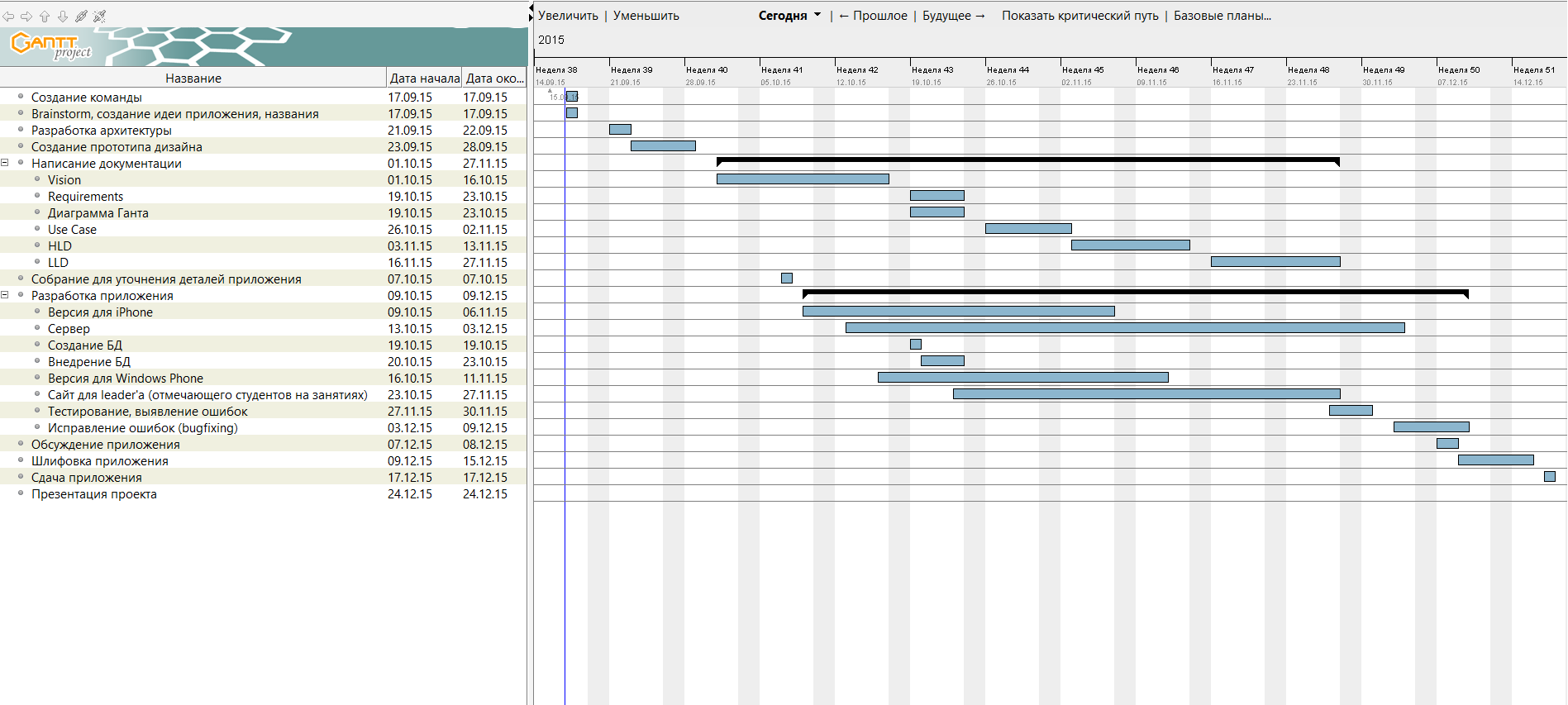
1. Ограничение времени: бета-тест и рекламная компания должны пройти в период начало декабря 2015 – конец января 2016. В продакшн проект должен выйти в начале 2 семестра учебы, то есть в начале февраля 2016 года.

2. Ограничение расходов: команда разработчиков работает за баллы и похвалу. Дальнейшее развитие приложение может идти за счет заинтересованности деканата нашего и других вузов.

3. Ограничение возможностей: проект ограничен разработкой, кастомизацией и реализацией. Дальнейшее развитие зависит от энтузиазма любых разработчиков, пожелавших работать с этим приложением, участвовать в его развитии и становлении на рынок.

# 4. Оценка ресурсов

Оценка целей и задач наглядно показана в приведенной ниже диаграмме Гантта:



# 5. Предположения и зависимости

Предположения:

1. Оповещения могут быть незамечены студентом, т.к. отображаются на экране мобильного устройства не постоянно.

2. Дизайн может быть слишком прост для искушенного пользователя iOS.

Зависимости:

1. Инструменты геолокации на iOS совсем не идеально определяют местоположение.

2. Через некоторый промежуток времени БД может быть «захламлена» от возрастающего количества студентов.