

Техническое задание на создание справочной системы «RecycleEasily»

Содержание

1. Введение
2. Основания для разработки
3. Назначение разработки
4. Требования к программе или программному изделию.
 - 4.1 Требования к функциональным характеристикам.
 - 4.1.1. Требования серверного приложения.
 - 4.1.2. Требования клиентского приложения.
 - 4.1.3. Пользовательские функции.
 - 4.2 Требования к надежности.
 - 4.3 Условия эксплуатации.
 - 4.4 Требования к составу и параметрам технических средств.
 - 4.5 Требования к информационной и программной совместимости.
 - 4.5.1 Серверное приложение.
 - 4.5.2 Клиентское приложение.
 - 4.6 Требования к маркировке и упаковке.
 - 4.7 Требования к транспортированию и хранению.
 - 4.8 Специальные требования.
5. Требования к программной документации.
6. Стадии и этапы разработки.
7. Порядок контроля и приемки.

1. Введение

Необходимо разработать справочную систему по утилизации и переработке вторичных материалов.

2. Основания для разработки

Основанием для разработки является учебный план кафедры вычислительной техники на 3-й семестр.

3. Назначение разработки

Помочь гражданам легко и быстро находить места, где можно сдать утиль на переработку, а так же получать информацию о том, как вести более экологически правильный образ жизни, популяризация экологически правильного образа жизни.

4. Требования к программе или программному изделию.

4.1 Требования к функциональным характеристикам.

4.1.1 Требования серверного приложения:

Серверное приложение должно отвечать следующим требованиям:

- обеспечение работы мобильных приложений системы
- реализация REST API для взаимодействия с мобильными приложениями
- формат сериализации данных JSON
- возможность двухсторонней синхронизации
- реализация системы администрирования
- серверная часть должна хранить базу данных пользователей, имеющих доступ к системе и обеспечивать аутентификацию пользователей согласно имеющимся записям.
- серверная часть должна хранить базу данных с пунктами утилизации и сведениями зарегистрированных пользователей.

4.1.2 Требования клиентского приложения:

Клиентское приложение должно отвечать следующим требованиям.

- Работать в операционных системах: Android, iOS.
- Использовать для соединения и обмена данными протокол TCP/IP.
- Иметь доступный и простой интерфейс пользователя.
- Иметь гибкую систему настроек.
- Клиентская часть должна хранить промежуточные данные для единовременной синхронизации с сервером.
- регистрация пользователя через e-mail или социальные сети.

4.1.3 Пользовательские функции:

В системе должны быть реализованы следующие пользовательские функции:

- вывод пунктов утилизации на карту
- фильтрация пунктов утилизации по параметрам
- справка по указанному объекту утилизации
- настройка своего профиля
- развитие профиля на основе совершенных действий
- автоматическая интеграция аккаунтов социальных сетей.

4.2 Требования к надежности.

Клиентское приложение должно работать в автономном режиме, без связи с серверным приложением. Обе части должны без потерь передавать информацию по каналу связи между клиентом и сервером.

4.3 Условия эксплуатации.

Стандартные условия эксплуатации программных продуктов.

Необходимые сотрудники для обслуживания серверной части системы – системный администратор для обслуживания собственно сервера и группа отвечающая за актуальность базы данных.

4.4 Требования к составу и параметрам технических средств.

Для нормальной работы серверной части необходимо:

- Сервер с тактовой частотой процессора не менее 500Mhz.
- Оперативная память не менее 512 Mb.
- Жесткий диск объемом не менее 5 Gb.
- Наличие адаптера подключения к сети (сетевой карты, модема и т.п.).
- Установленный дистрибутив Linux.
- Настроенный протокол TCP/IP.

Для работы клиентского приложения необходимо любое мобильное устройство с ОС Android или iOS.

4.5 Требования к информационной и программной совместимости.

4.5.1 Серверное приложение.

Серверная часть должна работать на любом дистрибутиве Linux с установленным интерпретатором языка программирования «Python», фреймворком «Django», системой управления базами данных «MySQL», а также всеми пакетами, необходимыми для работы выше описанного программного обеспечения. В качестве серверного ПО используется «Nginx» в связке с «UWSGI».

4.5.2 Клиентское приложение.

Клиентское приложение должно работать на операционных системах Android и iOS, дополнительного программного обеспечения не требуется. Для разработки приложения используются языки программирования C++, JavaScript, разрешается использовать открытые библиотеки программного обеспечения. Создание интерфейса разрешается в любом редакторе.

4.6 Требования к маркировке и упаковке.

Не предъявляются.

4.7 Требования к транспортированию и хранению.

Не предъявляются.

4.8 Специальные требования.

Не предъявляются.

5. Требования к программной документации.

Программной документацией к разрабатываемой системе является расчетно-пояснительная записка, также в клиентской части приложения необходимо создать раздел с инструкцией по использованию.

6. Стадии и этапы разработки.

№	Этап	Содержание работы	Срок	Исполнители
1	I	Создание базы данных пунктов утилизации.	01.09.16-15.09.16	Успенский А.Ф.
2		Настройка сервера и создание системы администрирования.	16.09.16-09.10.16	
3	II	Проектирование модулей клиентского приложения.	10.10.16-23.10.16	Группы: ИКБО-04-15 ИКБО-06-15
4		Реализация модулей клиентского приложения.	24.10.16-06.11.16	
5		Объединение модулей клиентского приложения.	07.11.16-20.11.16	
6	III	Тестирование всех частей приложения.	21.11.16-04.12.16	
7	IV	Составление документации приложения.	05.12.16-18.12.16	

7. Порядок контроля и приемки.

Испытание представленной системы и контроль качества ее работы провести

на базе случайных мобильных устройств, отвечающих требованиям п. 4.4. Во время испытаний проверить работу системы по следующим позициям:

- Запуск серверного и клиентского приложений.
- Соединение клиента (-ов) с сервером, проверка правильности обработки сервером соединения.
- Аутентификация пользователя на сервере и клиенте. Проверка изменения профиля зарегистрированных пользователей.
- Администрирование базы данных через серверное приложение.
- Проверка всех функций клиентской части.