**Лабораторная работа № 6**

**Тема:** Разработка отчетных форм для внедрения программных средств

**Цель:** разработка документации и отчетных форм.

“СберЦель”

**План установки ПО.**

**Перечень пользовательских мест, на которых должно быть установлено ПО:**

1. Мобильные устройства (смартфоны и планшеты) с операционной системой iOS (iPhone, iPad) и Android, которые будут использоваться для управления личными финансами и установки финансовых целей.
2. Личные компьютеры или ноутбуки с возможностью установки и использования мобильного приложения СберЦель через веб-версию или специализированное приложение.
3. Устройства с возможностью доступа в интернет, такие как умные часы, который могут поддерживать функции управления финансами через мобильное приложение СберЦель.

**Запланированные сроки установки ПО:**

Запланированные сроки установки ПО СберЦель зависят от конкретных потребностей и возможностей пользователя. Обычно установка мобильного приложения занимает всего несколько минут. Разрешение доступа и настройка приложения также могут быть завершены в течение нескольких минут. Для более сложных сценариев, таких как установка на несколько устройств или настройка множества целей и параметров управления, установка и настройка могут занять дополнительное время.

Обычно можно ожидать, что установка и настройка приложения СберЦель будут завершены в течение дня или даже быстрее, но это может быть предварительно спланировано на более длительный период, если необходимо установить на несколько устройств или провести обучение пользователей.

**Методы установки ПО:**

1. Через магазин приложений. Для устройств на базе iOS (iPhone, iPad) можно установить приложение СберЦель через App Store, где вам нужно просто найти приложение в магазине, нажать на кнопку "Установить" и следовать инструкциям.

2. Через веб-версию приложения. Пользователи также могут использовать веб-версию приложения, войдя на сайт Сбербанка через браузер на своем компьютере или мобильном устройстве. Затем они могут установить и настроить приложение через веб-интерфейс.

3. Через передачу файлов. В некоторых случаях, для установки на устройствах без доступа к магазину приложений, можно использовать передачу файлов через кабель, Bluetooth или другие методы передачи данных.

**Технические средства поддержки:**

1. Смартфон или планшет: устройство должно быть совместимо с операционной системой, поддерживаемой мобильным приложением "Сберцель".

2. Доступ к Интернету: для загрузки мобильного приложения "Сберцель" и получения обновлений необходим доступ к сети Интернет. Устройство должно быть оборудовано сотовой связью или иметь возможность подключения к Wi-Fi.

3. Пространство для хранения: мобильное приложение "Сберцель" занимает определенное пространство на устройстве, поэтому важно убедиться, что устройство имеет достаточное количество свободного пространства для установки приложения.

4. Инструкции и лицензии: необходимые документы, инструкции по установке и использованию мобильного приложения "Сберцель", а также информация о лицензировании приложения.

**Обучение персонала:**

1. Организация обучающего семинара или воркшопа, в рамках которого сотрудники будут получать подробную информацию о возможностях приложения, его основных функциях и способах использования.

2. Разработка обучающих материалов, таких как видеоуроки, инструкции, презентации и т.д., которые будут доступны сотрудникам для самостоятельного изучения.

3. Проведение практических занятий, в ходе которых сотрудники смогут попробовать использовать приложение на практике и получить обратную связь от опытных пользователей.

4. Предоставление консультаций и поддержки со стороны специалистов по работе с приложением, которые будут готовы отвечать на вопросы и помогать с решением возникающих проблем.

5. Проведение тестов или проверок знаний среди сотрудников для оценки уровня освоения материала.

**План передачи ПО**

**Краткий обзор системы и документов:**

1. Документация: включает технические спецификации, инструкции по установке и использованию, а также руководства пользователя, которые помогают пользователям и разработчикам понять функциональные возможности приложения.

2. Управление процессом разработки: система также включает в себя процессы управления разработкой, тестированием и внедрением новых версий приложения.

3. Сопровождение и поддержка: после выпуска приложения необходима система сопровождения и поддержки, включающая в себя обработку обратной связи пользователей, исправление багов, обновление безопасности и добавление новых функций.

**Детальное описание ресурсов, необходимых для поддержки передаваемого ПО:**

1. IT-специалисты: команда по поддержке приложения должна включать в себя IT-специалистов, способных реагировать на возникающие технические проблемы и проводить техническую поддержку для пользователей. Команда может включать в себя системных администраторов, разработчиков, инженеров поддержки и специалистов по безопасности.

2. Техническая инфраструктура: необходима надежная и масштабируемая инфраструктура, включающая в себя сервера, базы данных, средства мониторинга и управления работой приложения. Также важно иметь процессы для мониторинга и управления производительностью и доступностью приложения.

3. Обновления и патчи: необходимо обеспечить регулярные обновления приложения, включая исправления безопасности, устранение ошибок и добавление новых функций. Ресурсы должны быть выделены для разработки, тестирования и развертывания обновлений.

4. Техническая поддержка: центр поддержки пользователей должен быть обеспечен ресурсами для обработки запросов пользователей, решения технических проблем, обучения и консультирования пользователей.

5. Анализ данных и отчетность: необходимый ресурс для анализа данных о поведении пользователей, производительности приложения, обратной связи пользователей и других метрик, которые помогут в постоянном улучшении приложения.

**Перечень рекомендуемых мероприятий, в том числе консультации и лекции, которые должен проводить разработчик в целях поддержки передаваемого ПО и соответствующей среды поддержки:**

1. Организация консультаций по вопросам установки и настройки мобильного приложения сберцель.

2. Проведение лекций о принципах работы мобильного приложения сберцель и его взаимодействии с соответствующей средой поддержки.

3. Проведение обучающих мероприятий по использованию функционала мобильного приложения сберцель в соответствующей среде поддержки.

4. Предоставление консультаций по вопросам тестирования и отладки мобильного приложения сберцель в соответствующей среде.

5. Организация обучающих семинаров и практических занятий по оптимизации работы мобильного приложения сберцель в соответствующей среде поддержки.

**Описание проекта**

**Детализированное описание того, как ПО удовлетворяет специфицированным требованиям верхнего уровня к ПО:**

1. Алгоритмы: Описание используемых алгоритмов для основного функционала ПО сберцель, включая алгоритмы шифрования и безопасности передачи данных, а также алгоритмы аутентификации и авторизации пользователей.

2. Структуры данных: Описание структур данных, используемых для хранения информации о пользователях, транзакциях, и других необходимых данным, например, информации о банковских операциях.

3. Распределение по процессорам и задачам: Определение того, какие задачи и функции ПО сберцель могут быть распределены по процессорам и какие ресурсы им требуются, например, оптимизация процессов аутентификации и авторизации на серверной и клиентской стороне приложения.

4. Требования к безопасности: Описание механизмов обеспечения безопасности ПО сберцель, включая требования к шифрованию данных, защите от несанкционированного доступа и обработке конфиденциальной информации.

**Описание архитектуры ПО, которая определяет структуру ПО, предназначенного для реализации заданных требований:**

1. Многоуровневая архитектура: ПО Сберцель может быть организовано по принципу многоуровневой архитектуры, включая уровни бизнес-логики, уровни доступа к данным, пользовательский интерфейс и уровень интеграции с внешними системами.

2. Модульная структура: Приложение может быть построено на основе модульной структуры, разделяющей функционал на отдельные компоненты, каждый из которых отвечает за определенные задачи, например, аутентификацию, управление счетами, проведение транзакций, и т.д.

3. Использование микросервисов: Для повышения масштабируемости и гибкости, архитектура ПО Сберцель может быть построена на основе микросервисов, где отдельные функции приложения выносятся в отдельные сервисы, обеспечивающие их независимость и возможность разработки и поддержки.

4. Система безопасности: Архитектура ПО Сберцель должна включать систему безопасности, включая механизмы аутентификации, авторизации, шифрования данных, а также контроль доступа к различным функциям и данным приложения.

5. Интеграция с внешними системами: Архитектура ПО Сберцель должна предусматривать возможность интеграции с внешними системами, такими как банковские API, платежные системы и другие сторонние сервисы.

**Описание входных/выходных данных (например, словарь данных) для внутренних и внешних интерфейсов архитектуры ПО:**

1. Входные данные для внутренних интерфейсов:

- Данные пользователя (логин, пароль, профиль)

- Параметры поиска недвижимости (тип, район, цена, количество комнат и т.д.)

- Данные калькулятора ипотеки/рассрочки (сумма, процентная ставка, срок и др.)

- Данные из базы данных недвижимости (описание объектов, фотографии, планировки)

- Данные о счетах и накоплениях

2. Выходные данные для внутренних интерфейсов:

- Поисковые результаты (список объектов недвижимости соответствующих критериям)

- Детальная информация о недвижимости

- Результаты расчетов калькулятора ипотеки/рассрочки

3. Входные данные для внешних интерфейсов:

- Данные GPS для отображения объектов на карте

- Контактные данные для связи с агентами или собственниками

- Уведомления от операционной системы (пуши)

4. Выходные данные для внешних интерфейсов:

- Отображение геолокации объектов на карте

- Отправка запросов на связь с агентами или собственниками

- Отображение уведомлений для пользователей от операционной системы

**Описание потока данных и потока управления:**

1. Поток данных поиска недвижимости:

- Пользователь вводит критерии поиска (тип недвижимости, район, цена, количество комнат и т.д.).

- Приложение передает эти критерии серверу.

- Сервер обрабатывает запрос, выполняет поиск в базе данных и возвращает результаты поиска в виде списка объектов недвижимости.

2. Поток данных отображения детальной информации о недвижимости:

- Пользователь выбирает конкретный объект из списка.

- Приложение отправляет запрос на сервер для получения подробной информации о выбранном объекте.

- Сервер возвращает данные о недвижимости, включая фотографии, планировки, описание.

3. Поток данных калькулятора ипотеки/рассрочки:

- Пользователь вводит параметры кредита/рассрочки в соответствующую форму.

- Приложение отправляет данные на сервер для расчета ежемесячных платежей и общей суммы.

- Сервер возвращает результаты расчетов, которые отображаются пользователю.

**Поток управления включает в себя обработку действий пользователя и управление последовательностью отображаемых экранов в мобильном приложении:**

- Пользователь выбирает действие (например, поиск недвижимости, расчет ипотеки).

- Приложение обрабатывает действие пользователя и выполняет соответствующий запрос к серверу или базе данных.

- Приложение отображает результаты запроса на экране смартфона, предоставляя пользователю необходимую информацию.

**Методы разбиения ПО и средства обеспечения целостности при разбиении:**

1. Методы разбиения ПО:

- Функциональное разбиение: разделение программного обеспечения на функциональные модули или компоненты, которые выполняют определенные задачи или функции.

- Разбиение по уровням: разделение программного обеспечения на уровни, каждый из которых отвечает за определенные аспекты функционирования системы, например, уровень интерфейса пользователя, уровень бизнес-логики и т. д.

- Разбиение по объектам: разделение программного обеспечения на объекты, которые обладают определенными свойствами и выполняют определенные действия, таким образом, система разбивается на отдельные объекты, между которыми взаимодействует.

2. Средства обеспечения целостности при разбиении:

- Интерфейсы и контракты: определение четких интерфейсов и контрактов между различными компонентами программного обеспечения, чтобы обеспечить их совместную работу и интеграцию.

- Тестирование и проверка: проведение тестирования и проверки каждого компонента программного обеспечения, чтобы убедиться в его правильной работе и соответствии требованиям.

- Управление зависимостями: контроль зависимостей между компонентами программного обеспечения, чтобы предотвратить конфликты и непредвиденные последствия при изменениях в одном из компонентов.

- Использование стандартов и методологий: применение стандартных подходов и методологий разработки программного обеспечения, чтобы обеспечить единообразие и целостность системы при разбиении.

**Производные требования, полученные в процессе проектирования ПО:**

1. Требования к интерфейсу пользователя: это могут быть требования к дизайну и удобству использования приложения, включая адаптивный и интуитивно понятный интерфейс, поддержку жестов, оптимизацию под различные размеры экранов и удобное расположение элементов управления.

2. Требования к безопасности: это включает в себя обеспечение защиты данных пользователей, использование надежных методов аутентификации и авторизации, шифрование информации, защиту от вредоносных атак, а также соблюдение всех соответствующих законов о защите данных.

3. Требования к производительности: это включает в себя обеспечение быстрой загрузки и отклика приложения, оптимизацию использования ресурсов устройства (например, батареи, процессора, памяти) и минимизацию использования сетевого трафика.

4. Требования к мобильным платформам: это могут быть требования к поддержке определенных операционных систем (iOS, Android), версий ПО, устройств и разрешений экранов.

5. Требования к интеграции: это могут быть требования к интеграции с другими системами, такими как системы платежей, уведомлений, аналитики, а также социальных сетей.

6. Требования к локализации: это могут быть требования к поддержке различных языков, форматирования валюты, времени и культурных особенностей для удовлетворения потребностей местных пользователей.

**Для отключенного кода описание средств, гарантирующих невозможность его непреднамеренного выполнения:**

1. Отключение доступа: Код, который не должен выполняться, может быть отключен с помощью специальных флагов, условий или конструкций, которые запрещают его запуск. Таким образом, даже если код остается в системе, он не будет выполняться в процессе работы программного обеспечения.

2. Контроль версий: Отключенный код может быть переведен в состояние комментария или выведен в отдельную ветку кода, которая не входит в основную ветку разработки. В версионном управлении это позволяет отслеживать и контролировать код, который временно не используется, но остается доступным для будущих изменений или восстановления.

3. Документирование: Отключенный код должен быть ясно помечен и задокументирован, чтобы разработчики и другие заинтересованные лица знали, что он отключен намеренно и не должен выполняться. Это помогает избежать случайных попыток включить или использовать такой код без необходимости.

4. Проверки и тестирование: В ходе процесса разработки и тестирования программного обеспечения, следует проводить проверки и тесты, чтобы убедиться, что отключенный код действительно не выполняется и не влияет на работу системы. Это позволяет обнаружить возможные ошибки или уязвимости, связанные с отключенным кодом.

5. Регулярное обновление: Важно периодически рецензировать и обновлять проект программного обеспечения. Если отключенный код больше не требуется и не предполагается будущее его использование, он должен быть безопасно удален из системы, чтобы предотвратить потенциальные уязвимости или проблемы.

**Обоснование тех решений проекта, которые относятся к требованиям, связанным с безопасностью системы:**

1. Парольная защита: Для обеспечения безопасности пользовательских данных мобильное приложение сберцель должно иметь механизм парольной защиты. Пользователи должны создавать уникальные пароли, которые будут использоваться для входа в приложение.

2. Двухфакторная аутентификация: Для дополнительного уровня безопасности системы следует использовать двухфакторную аутентификацию. Это позволит убедиться, что только авторизованные пользователи имеют доступ к приложению.

3. Шифрование данных: Все пользовательские данные, передаваемые через мобильное приложение сберцель, должны быть зашифрованы, чтобы предотвратить их несанкционированный доступ.

4. Защита от вредоносного ПО: Мобильное приложение должно быть защищено от вредоносного программного обеспечения, поэтому следует использовать современные методы антивирусной защиты.

5. Обновление безопасности: Разработчики должны регулярно выпускать обновления приложения для устранения уязвимостей и повышения общей безопасности системы.

6. Система мониторинга безопасности: Для быстрого обнаружения и реагирования на потенциальные угрозы безопасности следует реализовать систему мониторинга, которая будет отслеживать аномальную активность и предупреждать об атаках.

7. Обучение пользователей: Кроме технических средств защиты, важно обучать пользователей правилам безопасного использования мобильного приложения, чтобы предотвратить утечку данных и другие инциденты.