

Министерство образования Республики Беларусь

Учреждение образования  
БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ  
ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Факультет компьютерных систем и сетей

Кафедра программного обеспечения информационных технологий

Дисциплина: Проектирование и разработка информационных систем

Практическая работа №1

Этапы проектирования и разработки ИС при структурном подходе к  
программированию. Стадия «Техническое задание»

Выполнили:  
студенты группы 051006:

Шуляк А. В.,  
Дранкевич А. А.

Проверила:

Грибович А. А.

Минск 2024

## СОДЕРЖАНИЕ

Введение.....	3
1 Наименование и область применения.....	4
2 Назначение разработки.....	5
3 Технические требования к программе.....	6
3.1 Требования к функциональным характеристикам.....	6
3.2 Требования к надёжности.....	6
3.3 Условия эксплуатации.....	7
3.4 Требования к информационной и программной совместимости.....	7
4 Стадии и этапы разработки.....	8
4.1 Стадии разработки.....	8
4.2 Этапы разработки.....	8
4.3 Содержание работ и временные ограничения.....	8
5 Порядок контроля и приемки.....	10

## ВВЕДЕНИЕ

Целью данного технического задания является создание программного решения для банковского приложения, нацеленного на автоматизацию операций и улучшение обслуживания клиентов. Продукт предназначен для использования сотрудниками банка с целью оптимизации процессов, снижения вероятности ошибок и повышения общей эффективности.

Банки представляют собой финансово-кредитные организации, осуществляющие широкий спектр операций с деньгами и ценными бумагами, предоставляющие финансовые услуги юридическим и физическим лицам. Они играют ключевую роль в современной финансовой системе, обеспечивая безопасность денежных транзакций, хранение средств, выдачу кредитов и другие услуги.

Исторически учёт клиентов и их счетов в банках велся на бумажных носителях, что сопряжено с определенными недостатками, такими как риск утери, невозможность автоматизированной проверки данных и значительные объемы документации. Однако в настоящее время доступны электронные системы, которые позволяют сократить влияние данных ограничений.

Исходя из вышесказанного, предполагается разработать веб-приложение для управления клиентской базой и счетами. Использование веб-технологий позволит эффективно распределить нагрузку между клиентскими и серверными устройствами. В качестве основной платформы выбран Python 3.12.

## **1 НАИМЕНОВАНИЕ И ОБЛАСТЬ ПРИМЕНЕНИЯ**

Наименование программного продукта - "XBank@". "XBank@" будет использоваться в банковской сфере для автоматизации различных банковских операций, включая управление счетами, обработку транзакций и обслуживание клиентов.

Основание для разработки

## **2 ОСНОВАНИЕ ДЛЯ РАЗРАБОТКИ**

Документ, на основании которого ведется разработка: Приказ №123 от 1 января 2024 года "О разработке программного продукта для банковского приложения".

Организация, утвердившая этот документ, и дата его утверждения: Банк "XBank", утверждено 2 января 2024 года.

Наименование и (или) условное обозначение темы разработки: Разработка программного продукта "XBank@" для автоматизации банковских операций.

### **3 НАЗНАЧЕНИЕ РАЗРАБОТКИ**

Функциональное назначение программы: Разработка программного продукта "XBank@" имеет целью автоматизацию различных банковских операций, таких как управление счетами, обработка транзакций и обслуживание клиентов. Программа будет предоставлять сотрудникам банка инструменты и функциональность для эффективного выполнения банковских операций, улучшения обслуживания клиентов и снижения ошибок.

Эксплуатационное назначение программы: Программа "XBank@" будет использоваться в банковской сфере для повышения эффективности работы банка, улучшения обслуживания клиентов и снижения затрат, связанных с ручной обработкой банковских операций. Программа будет предназначена для использования сотрудниками банка на рабочих станциях, но также будет совместима с различными операционными системами, такими как Windows, macOS и Linux.

## **4 ТЕХНИЧЕСКИЕ ТРЕБОВАНИЯ К ПРОГРАММЕ**

### **4.1 Требования к функциональным характеристикам**

Состав выполняемых функций:

а) Управление счетами:

- Открытие новых счетов с указанием необходимых данных клиента;
- Закрытие счетов с проверкой наличия задолженности или активных транзакций;
- Управление балансами счетов, включая пополнение и списание средств;

б) Обработка транзакций:

- Проведение вкладов на счета клиентов;
- Выплата наличных со счетов клиентов;
- Перевод средств между счетами клиентов внутри банка;
- Осуществление платежей на внешние счета или услуги;

в) Обслуживание клиентов

- Предоставление информации о счетах клиентов и их балансах;
- Ответы на вопросы клиентов с использованием функции чата или предоставление часто задаваемых вопросов;
- Предоставление отчетов о транзакциях клиентов и их счетах;

г) редактирование информации о клиенте банка;

д) проведение автоматизированной процедуры «Закрытие банковского дня»;

Организация входных и выходных данных;

а) Входные данные:

- Данные клиентов для открытия новых счетов или выполнения транзакций
- Данные запросов и вопросов клиентов для обслуживания и генерации ответов

б) Выходные данные

- Результаты операций: подтверждение открытия/закрытия счетов, подтверждение проведения транзакций
- Отчеты о транзакциях и состоянии счетов клиентов
- Ответы на запросы и вопросы клиентов

Временные характеристики;

а) Быстрая обработка транзакций и операций, чтобы минимизировать время ожидания клиентов и сотрудников банка:

б) Мгновенное отображение изменений балансов счетов после проведения транзакций:

- в) Высокая доступность системы без простоев и задержек
- г) Регулярное резервное копирование данных для обеспечения безопасности и возможности восстановления

Другие требования:

- а) Интерфейс программы должен быть интуитивно понятным и простым для использования сотрудниками банка
- б) Возможность масштабирования системы для поддержки роста числа клиентов и операций
- в) Безопасное хранение и передача данных, включая шифрование и механизмы аутентификации
- г) Контроль доступа к функциям и данным в соответствии с ролями и правами сотрудников банка

## **4.2 Требования к надёжности**

### **4.2.1 Обеспечение устойчивого функционирования**

- Программа должна иметь MTBF (среднее время безотказной работы) не менее 10 000 часов;
- Время восстановления после отказа должно быть не более 30 минут;
- При возникновении ошибок, программа должна генерировать подробные отчеты об ошибке для последующего анализа и устранения проблемы.

### **4.2.2 Контроль входной и выходной информации**

- Входные данные должны проходить проверку на валидность и целостность;
- Выходные данные должны быть проверены на соответствие заданным форматам и структурам данных;
- При обнаружении ошибок или несоответствий входных или выходных данных, программа должна предоставлять четкие сообщения об ошибке.

### **4.2.3 Время восстановления после отказа**

- Восстановление после отказа должно быть автоматическим и не превышать 15 минут;
- При восстановлении программы должны сохраняться все незавершенные операции и данные.

### **4.2.4 Обеспечение безопасности данных**

- Данные должны быть защищены с использованием современных методов шифрования, соответствующих стандартам безопасности



- Должны быть применены механизмы аутентификации и авторизации для доступа к данным
- Резервное копирование данных должно проводиться регулярно с возможностью восстановления данных за последние 24 часа

#### **4.2.5 Мониторинг и журналирование**

- Программа должна осуществлять постоянный мониторинг своего состояния и производительности
- Журналы операций и событий должны быть подробными и содержать достаточно информации для анализа проблем и выявления возможных нарушений безопасности.

#### **4.2.6 Тестирование и верификация**

- Программа должна пройти полное цикл тестирования, включая модульное, интеграционное и системное тестирование
- Верификация программы должна быть проведена независимой стороной для проверки соответствия требованиям и надежности работы

#### **4.2.7 Резервирование и отказоустойчивость**

- Должны быть предусмотрены механизмы резервирования, включая репликацию данных на отдельные сервера или хранилища

#### **4.2.8 Документирование и поддержка**

- Должна быть предоставлена подробная документация, описывающая архитектуру программы, процессы восстановления после отказа и инструкции по эксплуатации.
- Поддержка программы должна быть доступна посредством системы тикетов или электронной почты с гарантированными сроками ответа и решения проблем

### **4.3 Условия эксплуатации**

#### **4.3.1 Климатические условия эксплуатации**

Климатические условия нормальной работы программы соответствуют условиям нормальной работы оборудования, на котором будет использоваться программа, а также линий связи между единицами этого оборудования.

#### **4.3.2 Количество и квалификация персонала**

- Для обслуживания программы необходимо минимум два высококвалифицированных системных администратора с опытом работы в сфере информационных технологий не менее 5 лет
- Персонал должен обладать знаниями и навыками в области системного администрирования, баз данных, сетевых технологий и информационной безопасности

#### **4.3.3 Требования к составу и параметрам технических средств**

Для корректного функционирования программы требуется x64-совместимый персональный компьютер, включающий в себя:

- центральный процессор AMD EPYC 7H12 или выше;
- 128 ГБ ОЗУ или больше;
- 2 ТБ дискового пространства или больше;
- ОС Debian GNU/Linux 10.

Для получения доступа к приложению клиентские устройства должны обладать дисплеем с разрешением 1280x720 точек или выше.

#### **4.3.4 Вид обслуживания**

- Регулярное техническое обслуживание должно проводиться каждые 6 месяцев, включая проверку состояния носителей данных, чистку системного блока от пыли, обновление программного обеспечения и проведение тестов производительности.
- Мониторинг работы системы должен осуществляться в режиме реального времени с использованием специальных инструментов для обнаружения и предотвращения возможных проблем.

### **4.4 Требования к составу и параметрам технических средств**

#### **4.4.1 Серверы и вычислительное оборудование**

- Количество серверов: минимум 3 сервера для обеспечения отказоустойчивости.
- Процессоры: Многоядерные процессоры с тактовой частотой не менее 2.5 ГГц.
- Оперативная память (RAM): Минимум 32 ГБ RAM на каждом сервере.
- Жесткий диск: SSD с емкостью не менее 500 ГБ и скоростью чтения/записи не менее 500 МБ/с.

#### **4.4.2 Сетевое оборудование:**

- Маршрутизаторы/коммутаторы: Поддержка Gigabit Ethernet и возможность масштабирования сети.
- Пропускная способность: Минимум 1 Гбит/с для внешнего подключения и 10 Гбит/с для внутренней сети.

#### **4.4.3 Хранилище данных:**

- Объем хранилища: Минимум 10 ТБ емкости для хранения данных программы.
- Жесткий диск: RAID-массив из нескольких SSD-дисков с емкостью 2 ТБ каждый и скоростью чтения-записи не менее 1 ГБ/с.

#### **4.4.4 Резервные системы**

- Резервные серверы: Такие же характеристики, как и основные серверы, для обеспечения непрерывной работы.

#### **4.4.5 Устройства безопасности:**

- Шифрование данных: Использование симметричного алгоритма AES-256 для шифрования данных.
- Устройства обнаружения вторжений: Установка сетевых межсетевых экранов и системы обнаружения вторжений (IDS) с поддержкой сигнатурных и аномальных методов обнаружения.

#### **4.4.6 Мониторинг и управление:**

- Система мониторинга: Настройка инструментов мониторинга, таких как Nagios или Zabbix, для отслеживания состояния серверов, сети и приложений.

#### **4.4.7 Конечные компьютеры-терминалы доступа для программы:**

- Процессор: Многоядерный процессор с тактовой частотой не менее 2.0 ГГц. Оперативная память (RAM): Минимум 8 ГБ RAM для обеспечения плавной работы приложений.
- Жесткий диск: SSD с емкостью не менее 256 ГБ для быстрой загрузки операционной системы и приложений.
- Графический адаптер: Встроенный или дискретный графический адаптер.
- Операционная система: Windows 10, macOS либо ArchLinux с поддержкой требуемых программных компонентов.
- Сетевой интерфейс: Поддержка Ethernet для подключения к сети.
- Монитор: Минимум 17-дюймовый монитор с разрешением не менее 1280x1024 пикселей.

- Клавиатура и мышь: Стандартные устройства ввода для управления приложениями.

## **4.5 Требования к информационной и программной совместимости**

### **4.5.1 Информационные структуры на входе и выходе:**

- Форматы данных: Входные данные должны быть представлены в формате XML, версия 1.0, версия 1.0. Выходные данные должны быть в формате JSON.

### **4.5.2 Методы решения и исходные коды:**

- Язык программирования: Исходный код программы должен быть написан на языке Python версии 3.9 с использованием фреймворка Django
- Алгоритмы: Для решения конкретных задач используется алгоритм сортировки QuickSort и алгоритм поиска в ширину (BFS)

### **4.5.3 Защита информации и программ**

- Шифрование данных: Используется протокол SSL/TLS с алгоритмом шифрования AES-256 для защиты передаваемой информации
- Аутентификация и авторизация: Пользователи должны проходить аутентификацию с использованием уникальных учетных данных. Должна быть реализована система авторизации с разграничением доступа на основе ролей и прав доступа
- Защита от вредоносного кода: Программа должна включать механизмы защиты от вредоносного кода, такие как проверка наличия и обновление антивирусного программного обеспечения и брандмауэра

## **4.6 Требования к маркировке и упаковке**

### **4.6.1 Маркировка программного изделия**

- Логотип и наименование: Программное изделие должно иметь ясно видимый логотип и наименование компании или бренда, размером не менее 2 см x 2 см, на обложке упаковки.
- Версия и номер лицензии: На обложке программного изделия должны быть указаны версия программы и номер лицензии в виде текста, шрифтом не меньше 10 пунктов.
- Информация об авторских правах: На обложке программного изделия должна быть приведена информация об авторских правах в формате "© Год Владелец прав. Все права защищены"

#### **4.6.2 Упаковка программного изделия**

- Формат упаковки: Упаковка должна быть в виде картонной коробки с размерами 20 см x 15 см x 3 см.
- Материалы упаковки: Коробка должна быть изготовлена из прочного картона, толщиной не менее 2 мм, с глянцевой поверхностью.
- Дополнительные компоненты: В упаковку должны входить лицензионный ключ на отдельном бумажном носителе и информационный лист с инструкцией по установке и активации программы.

#### **4.6.3 Носитель ПО**

- Флеш-накопитель: Программное обеспечение должно быть упаковано на USB-флеш-накопителе с емкостью не менее 8 ГБ. Накопитель должен быть безопасно закреплен внутри упаковки, чтобы предотвратить его повреждение или потерю во время транспортировки и хранения. Флеш-накопитель должен быть маркирован с указанием логотипа и наименования компании или бренда программного изделия.

### **4.7 Требования к транспортировке и хранению**

#### **4.7.1 Требования к транспортированию**

- Упаковка: Программное изделие должно быть упаковано в прочную и защищенную упаковку, которая обеспечивает достаточную амортизацию и защиту от ударов и вибрации во время транспортировки.
- Защита от пыли и влаги: Упаковка должна быть пыле- и влагозащищенной, чтобы предотвратить попадание влаги или пыли на программный носитель

#### **4.7.2 Требования к месту хранения**

- Температура и влажность: Программное изделие должно храниться в сухом и прохладном помещении с контролируемой температурой в диапазоне от +5°C до +25°C и относительной влажностью не более 60%.
- Защита от воздействия окружающей среды: Программное изделие должно быть защищено от прямого солнечного света, вибрации, сильных магнитных полей и других внешних воздействий, которые могут повредить носитель ПО.

#### **4.7.3 Требования к условиям хранения**

- Положение и укладка: Программные носители должны быть хранены в вертикальном положении без нагрузки сверху, чтобы предотвратить их деформацию или повреждение.
- Защита от статического электричества: При хранении программных носителей необходимо предотвращать статическое электричество, поэтому рекомендуется использовать антистатические упаковочные материалы или контейнеры.

#### **4.7.4 Условия складирования**

- Складские условия: Программное изделие должно храниться в специально отведенном складском помещении, свободном от воздействия влаги, пыли, агрессивных химических веществ и других факторов, которые могут повредить носитель ПО.
- Упаковка и маркировка: Программное изделие должно быть хранено в оригинальной упаковке с ясно видимой маркировкой, указывающей на условия хранения и предупреждающей о необходимости бережного обращения.

#### **4.7.5 Сроки хранения в различных условиях**

- Рекомендуемый срок хранения: Программное изделие должно иметь указанный рекомендуемый срок хранения, который может быть указан на упаковке или в прилагаемой документации.
- Условия экстремальной температуры или влажности: Если программное изделие может быть подвержено экстремальным температурам или влажности во время хранения, должны быть указаны соответствующие сроки хранения или рекомендации для таких условий.

## 5 ТЕХНИКО-ЭКОНОМИЧЕСКИЕ ПОКАЗАТЕЛИ

### 5.1.1 Ориентировочная экономическая эффективность

- Расчет ожидаемой прибыли: Предполагаемая цена продажи программного изделия составляет \$100 за лицензию, а планируется продать 10 000 лицензий в год. Ожидаемая годовая прибыль составляет \$1 000 000.
- Расчет возвратности инвестиций: Затраты на разработку и производство программного изделия составляют \$500 000. При годовой прибыли в размере \$1 000 000, срок окупаемости составляет полгода.
- Оценка рентабельности: Годовая прибыль составляет \$1 000 000, а затраты на разработку и производство программного изделия составляют \$500 000. Рентабельность равна 100%  $((\$1\,000\,000 - \$500\,000) / \$500\,000)$ .

### 5.1.2 Предполагаемая годовая потребность

- Оценка рыночного спроса: Исследования показывают, что годовая потребность в программном изделии составляет около 20 000 лицензий.
- Идентификация целевой аудитории: Программное изделие предназначено для бизнес-клиентов в сфере финансового управления, и целевая аудитория охватывает средние и крупные предприятия в финансовом секторе.

### 5.1.3 Экономические преимущества разработки

- Сравнение с конкурентами: "XBank@" предлагает более эффективный и удобный интерфейс по сравнению с основными конкурентами, что позволяет сократить время обучения пользователя и повысить производительность на 20%.
- Уникальные особенности: "XBank@" обладает интегрированным алгоритмом искусственного интеллекта, который автоматизирует сложные финансовые анализы и снижает риски на 30%

## **6 СТАДИИ И ЭТАПЫ РАЗРАБОТКИ**

### **6.1 Стадии разработки**

– Планирование

i) Разработка технического задания (ТЗ)

ii) Согласование и утверждение ТЗ

iii) Исполнитель: Бизнес-аналитик

– Проектирование

i) Этап 1: Архитектурное проектирование

(1) Разработка архитектуры программного изделия

(2) Согласование и утверждение архитектуры

(3) Исполнитель: Системный архитектор

ii) Этап 2: Проектирование пользовательского интерфейса

(1) Разработка макетов и прототипов интерфейса

(2) Согласование и утверждение дизайна интерфейса

(3) Исполнитель: Дизайнер пользовательского интерфейса

– Разработка

i) Этап 1: Написание кода и тестирование

(1) Разработка программного кода

(2) Проведение модульного тестирования

(3) Исполнитель: Команда разработчиков

ii) Этап 2: Интеграция и системное тестирование

(1) Интеграция различных модулей программы

(2) Проведение системного тестирования

(3) Исполнитель: Команда разработчиков

– Внедрение и поддержка

i) Этап 1: Установка и настройка программного изделия

(1) Установка и настройка программы на клиентских системах

(2) Проведение тестирования в реальных условиях

(3) Исполнитель: Команда технической поддержки

ii) Этап 2: Поддержка и обновление

(1) Регулярное обслуживание и обновление программного изделия

(2) Решение проблем и предоставление технической поддержки

(3) Исполнитель: Команда технической поддержки

### **6.2 Содержание работ и временные ограничения**

Содержание работ и временные ограничения приведены на диаграмме Ганта, отображённой на рисунке 6.2.1.



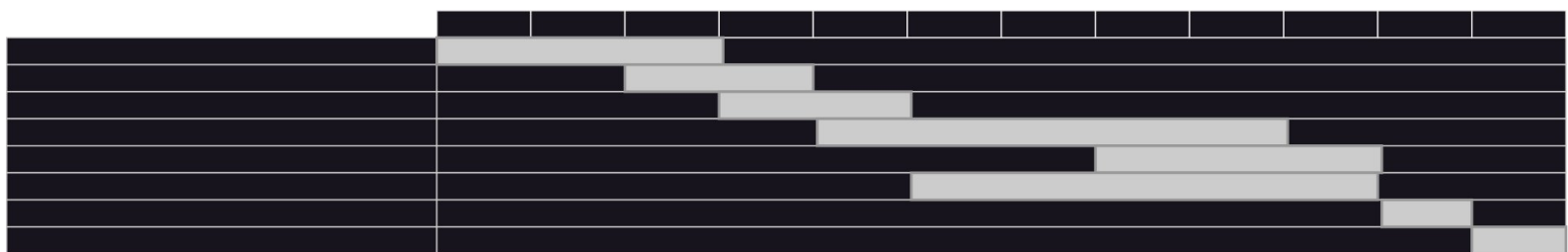


Рисунок 6.2.1 – Диаграмма Ганта содержания работ и сроков разработки

## **7 ПОРЯДОК КОНТРОЛЯ И ПРИЕМКИ**

### **7.1 Виды испытаний**

- Модульное тестирование: Проверка отдельных модулей программы на соответствие требованиям и ожидаемому поведению.
- Системное тестирование: Проверка работы программы в целом, включая взаимодействие между компонентами и выполнение функциональных требований.
- Интеграционное тестирование: Проверка взаимодействия между различными модулями и компонентами программы.
- Пользовательское тестирование: Проведение тестирования программы со стороны конечных пользователей для проверки удобства использования, функциональности и соответствия требованиям.
- Нагрузочное тестирование: Проверка работоспособности и производительности программы при высоких нагрузках и большом объеме данных.

### **7.2 Общие требования к приемке работы**

- Работа должна соответствовать утвержденному техническому заданию и спецификациям.
- Все функциональные требования должны быть реализованы и работать корректно.
- Работоспособность программы должна быть проверена и подтверждена через проведение испытаний.
- Программа должна быть документирована соответствующим образом, включая руководство пользователя и техническую документацию.
- Приемка работы должна быть проведена заказчиком или уполномоченной им стороной с участием представителей разработчиков.

## **8 ПРИЛОЖЕНИЯ**

Перечень научно-исследовательских и других работ, обосновывающих разработку:

- Отчеты о проведенных исследованиях рынка и конкурентной среды, включая анализ требований пользователей и особенностей аналогичных программных продуктов.
- Экспертные заключения и консультации специалистов в области, подтверждающие необходимость разработки программного изделия.
- Протоколы и результаты проведенных исследовательских работ, лежащих в основе технологических решений, применяемых в программном изделии