

JARVIS TECHEXPENSE:THE EXPENSE TRACKER

SOFTWARE ENGINEERING AND CONCEPTS

DOCUMENTATION



Submitted by

JITHEESWARAN V	220701108
KIRUTHHIK A S	220701131
JYOTHI SAKTHI H	220701113
AKASH M	220701502
MANOJ KUMAR J	220701524
KARTHIGA R	220701119
JEYSRI PR	220701107

In partial fulfillment for the award of the degree of

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE

RAJALAKSHMI ENGINEERING COLLEGE (AUTONOMOUS)

THANDALAM

CHENNAI-602105

2024 - 25

TABLE OF CONTENTS

1. INTRODUCTION

1.1 INTRODUCTION

1.2 PROBLEM STATEMENT

2. BUSINESS ANALYSIS

2.1 BUSINESS ARCHITECTURE

2.2 HOW DIFFERENT FROM OTHERS APPLICATIONS

2.3 PROCESS WORKING SYSTEM

2.4 BUSINESS PROBLEM AND SOLUTION

3. REQUIREMENTS

3.1 IMPLEMENTATION OVERVIEW

3.2 EPIC

3.3 FEATURES

3.4 USER STORY

3.5 NON FUNCTIONAL REQUIREMENTS

3.6 POKER PLANNING

3.7 MODULES

4. ARCHITECTURE

4.1 OVERVIEW

4.2 CLASS DIAGRAM

4.3 SEQUENCE DIAGRAM

5. TEST STRATEGY

5.1 TEST PLANS

5.2 TEST CASE

6. DEPLOYMENT

6.1 DEPLOYMENT ARCHITECTURE

6.2 AZURE SERVICES USED

CHAPTER 1

INTRODUCTION

1.1 Introduction

In today's fast-paced financial landscape, effective expense management is crucial for both individuals and businesses. With increasing financial complexities and the need for meticulous tracking of expenditures, traditional methods often fall short. Enter JARVIS TechExpense, an innovative expense tracking solution designed to address these challenges head-on. This advanced platform offers a seamless and intuitive approach to monitoring, categorizing, and analyzing expenses in real-time. By integrating cutting-edge technology with a user-friendly interface, JARVIS TechExpense simplifies financial oversight, providing users with the tools they need to gain comprehensive insights into their spending habits, ensure accuracy, and achieve their financial goals. Whether for personal budgeting or managing business finances, JARVIS TechExpense sets a new standard in financial management, making it an essential tool in the modern financial toolkit.

1.2 Problem statement

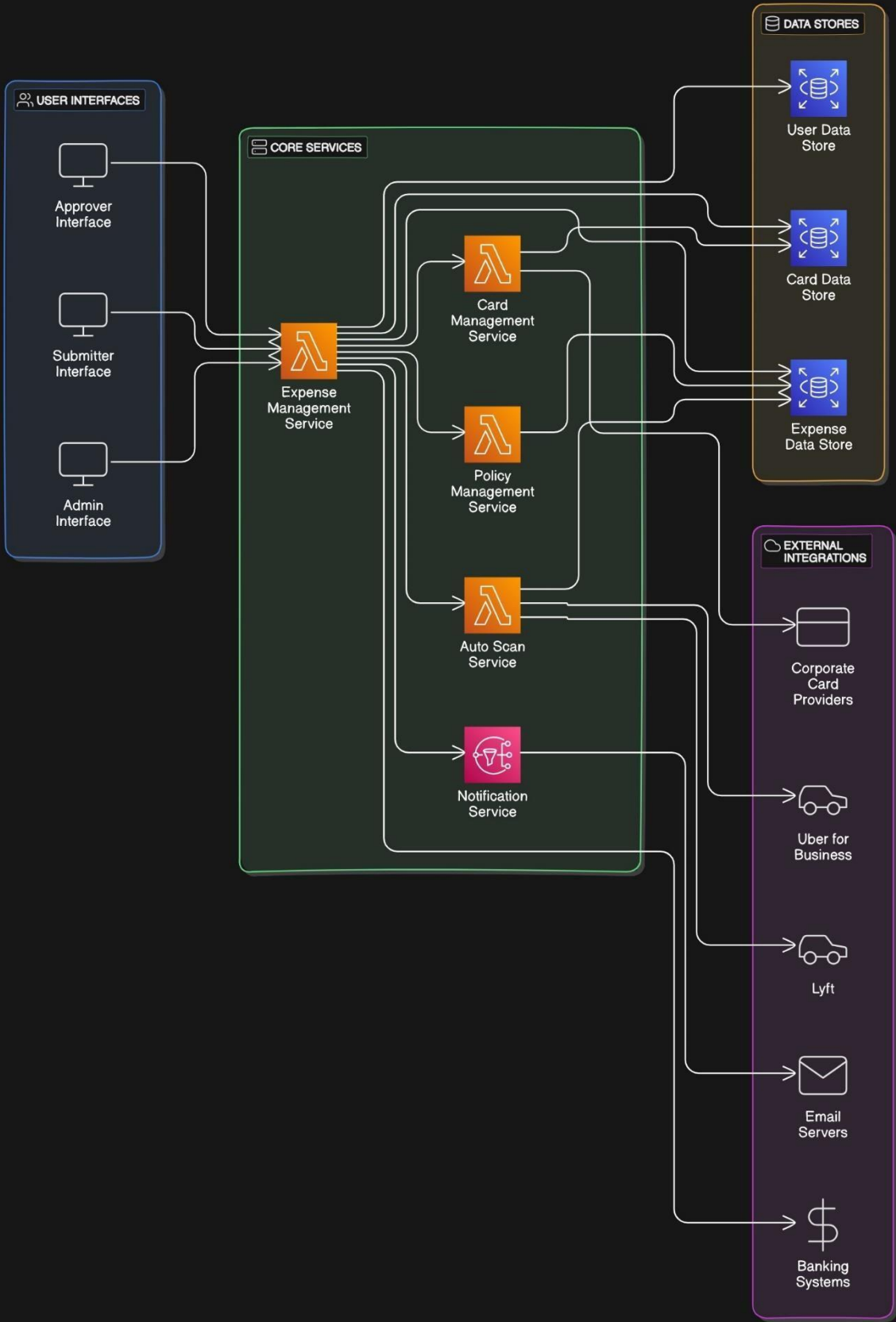
Managing personal and business finances has become increasingly complex in the digital age, with individuals and organizations facing numerous challenges in tracking and controlling their expenses. Traditional methods of expense management, such as manual entry and paper-based systems, are time-consuming, prone to errors, and lack real-time insights. These limitations hinder effective financial planning and decision-making, leading to potential overspending, missed savings opportunities, and inadequate financial oversight. There is a pressing need for a comprehensive, automated solution that can seamlessly integrate with various financial accounts, provide accurate and real-time tracking, and offer advanced analytics to help users understand their spending patterns and make informed financial decisions. JARVIS TechExpense aims to address these challenges by offering a robust, user-friendly expense tracking platform that enhances efficiency, accuracy, and financial awareness for both individuals and businesses.

Ch 2. BUSINESS ANALYSIS

2.1 BUSINESS ARCHITECTURE

Figure 1

Business Architecture Diagram



2.2 HOW DIFFERENT FROM OTHER APPLICATION

1. *Cost Efficiency*

- **Reduced Administrative Overhead:** By automating expense tracking and approval processes, JARVIS TechExpense significantly reduces the administrative burden, leading to lower operational costs.
- **Error Reduction:** Automation and advanced data entry technologies like OCR minimize human errors, leading to more accurate expense reporting and fewer costly mistakes.

2. *Scalability*

- **Flexible Deployment:** JARVIS TechExpense can scale with the growth of a business, accommodating increasing numbers of users and transactions without compromising performance.
- **Modular Architecture:** Businesses can customize the application by adding or removing modules as needed, ensuring they only pay for what they use.

3. *Improved Compliance and Audit Readiness*

- **Regulatory Compliance:** The application is designed to meet various regulatory requirements, helping businesses comply with financial regulations effortlessly.
- **Audit Trails:** Comprehensive audit trails ensure that all expense-related activities are logged and traceable, facilitating easier and more transparent audits.

4. *Enhanced Employee Productivity*

- **Streamlined Processes:** Automated workflows and real-time tracking free up employees from manual expense reporting tasks, allowing them to focus on more strategic activities.
- **Mobile Accessibility:** With mobile access, employees can manage their expenses on the go, reducing delays and ensuring timely submissions.

5. *Better Financial Control and Transparency*

- **Real-Time Insights:** Detailed reporting and analytics provide real-time insights into spending patterns, enabling better financial planning and control.
- **Budget Management:** Businesses can set budget limits and monitor expenses against these budgets, helping to prevent overspending and ensuring financial discipline.

6. *Enhanced Decision-Making*

- **Data-Driven Insights:** Advanced analytics and reporting tools enable businesses to make informed decisions based on accurate and comprehensive expense data.
- **Predictive Analysis:** The application's analytics capabilities can help forecast future expenses, aiding in budget preparation and financial planning.

7. *Improved User Experience*

- **Intuitive Interface:** An easy-to-use interface reduces the learning curve for employees, leading to higher adoption rates and more effective use of the application.
- **Personalized Dashboards:** Customizable dashboards allow users to tailor the interface to their specific needs, enhancing usability and satisfaction.

8. *Seamless Integration*

- **ERP and Accounting Systems:** JARVIS TechExpense integrates seamlessly with major ERP and accounting systems, ensuring smooth data flow and reducing the need for duplicate data entry.

- **APIs for Custom Integration:** Businesses with unique requirements can leverage API access to integrate JARVIS TechExpense with other internal systems.

9. Robust Security Measures

- **Data Protection:** The application uses state-of-the-art encryption and security protocols to protect sensitive financial data.
- **Access Control:** Role-based access control ensures that only authorized personnel can access specific data, enhancing security and compliance.

10. Customer Support and Training

- **24/7 Support:** Continuous customer support ensures that any issues are promptly addressed, minimizing downtime and disruptions.
- **Comprehensive Training:** Extensive training resources help employees quickly become proficient in using the application, ensuring a smooth implementation process.

2.3 BUSINESS PROBLEM AND SOLUTION

1. Inefficiency and High Administrative Costs

- **Problem:** Manual expense tracking and reporting processes are time-consuming and laborintensive, leading to high administrative costs and inefficiencies.
- **Solution:** JARVIS TechExpense automates these processes, reducing the time and effort required for data entry, approval, and reporting. This leads to lower operational costs and increased efficiency.

2. Inaccurate Expense Reporting and Data Entry Errors

- **Problem:** Manual entry of expense data is prone to errors, resulting in inaccurate financial records and potential compliance issues.
- **Solution:** The application utilizes OCR technology to automatically scan and input data from receipts, significantly reducing the risk of human error and ensuring more accurate reporting.

3. Lack of Real-Time Expense Tracking and Financial Control

- **Problem:** Businesses struggle to monitor expenses in real-time, leading to poor financial control and unexpected budget overruns.
- **Solution:** JARVIS TechExpense provides real-time tracking of expenses and detailed reporting, enabling businesses to monitor spending patterns, set budget limits, and prevent overspending.

4. Complex and Slow Approval Workflows

- **Problem:** Traditional approval processes are often slow and cumbersome, causing delays in expense reimbursement and employee dissatisfaction.
- **Solution:** The application offers customizable approval workflows and automated notifications, streamlining the approval process and ensuring timely reimbursements.

5. Difficulty in Compliance and Audit Readiness

- **Problem:** Ensuring compliance with financial regulations and preparing for audits can be challenging and resource-intensive.
- **Solution:** JARVIS TechExpense is designed to comply with various financial regulations and provides comprehensive audit trails, making it easier for businesses to maintain compliance and be audit-ready.

6. Inconsistent and Fragmented Data Across Systems

- **Problem:** Expense data is often fragmented across different systems, leading to inconsistencies and difficulties in financial analysis.
- **Solution:** The application integrates seamlessly with popular accounting and ERP systems, ensuring consistent and consolidated data across platforms.

7. Poor User Experience and Low Adoption Rates

- **Problem:** Many expense management tools have complex interfaces that hinder user adoption and satisfaction.
- **Solution:** JARVIS TechExpense features a user-friendly interface and customizable dashboards, enhancing user experience and promoting higher adoption rates among employees.

8. Insufficient Data Security Measures

- **Problem:** Protecting sensitive financial data from breaches and ensuring data integrity is a major concern for businesses.
- **Solution:** The application employs robust encryption techniques and role-based access control to safeguard data both in transit and at rest.

Ch 3. REQUIREMENTS

3.1 IMPLEMENTATION OVERVIEW

The implementation of JARVIS TechExpense involves developing a robust, scalable platform with a user-friendly interface and secure backend infrastructure. It integrates with financial institutions via secure APIs for automated transaction imports. Advanced analytics, customizable reporting, and budgeting tools with alerts help users manage finances effectively. Real-time multi-device synchronization is achieved through a cloud-based infrastructure, with robust encryption ensuring data security. Extensive user testing, followed by high-availability deployment and ongoing maintenance, ensures a reliable and secure expense tracking solution.

3.2 EPIC

- User Authentication and Access Control
- Expense Logging and Categorization
- Expense Splitting and Settlement
- Expense Analysis and Reporting
- Interactive Data Visualization(dashboard)
- Budget Management
- Integration with Payment Platforms

3.3 FEATURES

1. Expense Submission Form:

A form where submitters can enter details such as date, amount, category, description, and attachments (e.g., receipts) for new expenses.

2. Expense Status Dashboard:

A dashboard where submitters can track the status of their submitted expenses (e.g., pending, approved, rejected) in real-time.

3. Expense Review Dashboard:

A dashboard for approvers to review all submitted expenses, showing detailed information and providing options to approve, reject, or request more information.

4. Approval Workflow:

A workflow that allows approvers to change the status of expenses to approved, rejected, or needs more info, with notifications sent to submitters.

5. Expense Categories:

Predefined and customizable categories for users to select when submitting expenses, with an admin interface to manage these categories.

6. Report Generation Tool:

A tool to generate reports based on criteria such as date range, category, and user, with visualizations like charts and graphs.

7. Real-time Notifications:

Instant notifications (via email or app) to submitters when an expense is approved or rejected, and customizable notification preferences.

8. Secure User Authentication:

Secure login using username and password, with optional two-factor authentication (2FA) for added security.

9. Detailed Expense Entry Form:

A detailed form for submitters to log their expenses accurately, including options for recurring expenses and auto-fill for common fields.

10. Budget Setup and Alerts:

An interface to set budget limits for different categories, with notifications for submitters when they approach or exceed these limits, and a visual budget dashboard.

3.4 USERSTORY

1. Submit Expense:

User Story: As a submitter, I want to submit an expense for approval so that it can be reviewed and reimbursed.

2. Track Status:

User Story: As a submitter, I want to track the status of my expense submission so that I know when it has been approved or rejected.

3. Review Submitted Expenses:

User Story: As an approver, I want to review submitted expenses so that I can verify their validity.

4. Approve or Reject Expenses:

User Story: As an approver, I want to approve or reject expenses so that valid expenses can be reimbursed and invalid ones can be denied.

5. Categorize Expenses:

User Story: As a user, I want to categorize my expenses so that I can understand where my money is going.

6. Generate Report:

User Story: As a user, I want to generate a report so that I can analyse my spending over a period of time.

7. Submitter Notifications:

User Story: As a submitter, I want to receive notifications when my expense is approved or rejected so that I can stay updated on its status.

8. Secure Login:

User Story: As a submitter, I want to securely log in to the expense management system using my username and password, so that I can submit and manage my expenses conveniently.

9. Log Expenses:

User Story: As a submitter, I want to log my expenses by entering details such as date, amount, description, and category, so that I can accurately track my spending and submit for approval.

10. Set Budget Limits:

User Story: As a submitter, I want to set budget limits for different expense categories to manage my spending, and receive notifications when I approach or exceed these limits, enabling me to control my expenses effectively

3.5 NON-FUNCTIONAL REQUIREMENTS

1. Performance

- The system should load and display the dashboard within 2-8seconds. - Expense entry should be processed within 2-4second.

2. Scalability

- The system should support up to X users without performance degradation. - It should be able to handle up to Y simultaneous user sessions.

3. Security

- User data should be encrypted in transit (using HTTPS) and at rest. - The system should implement strong password policies and multi-factor authentication (MFA).

4. Usability

- The user interface should be easy to navigate for users of all technical levels.
- The mobile app should follow platform-specific design guidelines.

5. Availability

- The system should have an uptime of 99.99% per month.
- It should support automatic failover to a backup server in case of primary server failure.

6. Maintainability

- The system codebase should follow best practices for readability and modularity to facilitate easy maintenance and updates.
- Documentation for the system should be comprehensive and up-to-date.

7. Data Consistency

- The system should ensure data consistency across different platforms (web, mobile) in real-time.

8. Compliance

- The system should comply with relevant regulations such as GDPR for data protection.
- Financial transactions should comply with applicable financial regulations.

9. Localization

- The system should support multiple languages.
- Date and currency formats should be customizable based on the user's comfort.

10. Backup and Disaster Recovery

- The system should perform daily backups and retain them for at least 15 days.
- It should support a recovery time objective (RTO) of 1 hour and a recovery

3.6 POKER PLANNING

Poker Planning, also known as Planning Poker, is an estimation technique used in Agile project management to achieve consensus on the effort required for various tasks. Below, I've outlined the Poker Planning estimates for each user story associated with the JARVIS TechExpense application.

User Stories and Estimates

1. **Submit Expense:**

- **User Story:** As a submitter, I want to submit an expense for approval so that it can be reviewed and reimbursed.
- **Estimate:** 5 story points

2. **Track Status:** ○ **User Story:** As a submitter, I want to track the status of my expense submission so that I know when it has been approved or rejected.

- **Estimate:** 3 story points

3. **Review Submitted Expenses:**

- **User Story:** As an approver, I want to review submitted expenses so that I can verify their validity.
- **Estimate:** 5 story points

4. **Approve or Reject Expenses:** ○ **User Story:** As an approver, I want to approve or reject expenses so that valid expenses can be reimbursed and invalid ones can be denied.

- **Estimate:** 5 story points

5. **Categorize Expenses:** ○ **User Story:** As a user, I want to categorize my expenses so that I can understand where my money is going.

- **Estimate:** 3 story points

6. **Generate Report:** ○ **User Story:** As a user, I want to generate a report so that I can analyze my spending over a period of time.

- **Estimate:** 8 story points

7. **Submitter Notifications:** ○ **User Story:** As a submitter, I want to receive notifications when my expense is approved or rejected so that I can stay updated on its status.

- **Estimate:** 3 story points

8. **Secure Login:**

- **User Story:** As a submitter, I want to securely log in to the expense management system using my username and password, so that I can submit and manage my expenses conveniently.
- **Estimate:** 5 story points

9. **Log Expenses:**

- **User Story:** As a submitter, I want to log my expenses by entering details such as date, amount, description, and category, so that I can accurately track my spending and submit for approval.
- **Estimate:** 5 story points

10. **Set Budget Limits:**

- **User Story:** As a submitter, I want to set budget limits for different expense categories to manage my spending, and receive notifications when I approach or exceed these limits, enabling me to control my expenses effectively.

- **Estimate:** 8 story points

Summary of Estimates

User Story	Estimate (Story Points)
Submit Expense	5
Track Status	3
Review Submitted Expenses	5
Approve or Reject Expenses	5
Categorize Expenses	3
Generate Report	8
Submitter Notifications	3
Secure Login	5
Log Expenses	5
Set Budget Limits	8
Total	50

3.7 Modules

- OCR to recognize the bills
- Manual bills
- Dashboard
- Report monthly and daily
- Bank details for transaction details

Ch 4. ARCHITECTURE

4.1 OVERVIEW

User Authentication Module:

- Handles secure login and user management.
- Interacts with the Expense Submission, Expense Tracking, and Notification Modules.

Expense Submission Module:

- Allows users to log and submit expenses for approval.
- Communicates with the Expense Tracking and Review Modules.

Expense Tracking Module:

- Tracks the status of submitted expenses.
- Integrates with the Notification Module to alert users of status changes.

Expense Review Module:

- Enables approvers to review, approve, or reject expenses.
- Updates the Expense Tracking Module with the decision.

Reporting Module:

- Generates reports based on categorized expenses and user preferences.
- Retrieves data from the Data Storage Module.

Budget Management Module:

- Allows users to set budget limits and tracks spending against these limits.
- Sends alerts through the Notification Module when limits are approached or exceeded.

Notification Module:

- Manages sending notifications to users regarding expense approvals, rejections, and budget alerts.

Error Handling:

- Validation Errors: Ensures data entered by users meets required formats and constraints.
- Authentication Errors: Handles failed login attempts and provides appropriate feedback.
- Network Errors: Manages connectivity issues and retries operations as needed.
- Data Errors: Detects and resolves inconsistencies in data storage and retrieval.

Azure SQL Database:

Structured Data Storage: Utilizes Azure SQL Database for storing structured data, ensuring scalability, security, and high availability. The database supports relational data storage, making it suitable for the complex queries and transactions required by JARVIS TechExpense.

Logging:

Event Logging: Logs user actions such as submitting expenses and logging in. These logs are stored in dedicated tables within Azure SQL Database to track user interactions and activity.

Error Logging: Captures and logs errors for troubleshooting and debugging. Error logs are stored in Azure SQL Database, enabling easy access and analysis by the development and support teams.

Audit Logging: Tracks changes made to expense reports and approvals for compliance and auditing purposes. These audit logs are maintained in Azure SQL Database to ensure traceability and accountability.

Data Storage Features:

- **ACID Properties:** Ensures Atomicity, Consistency, Isolation, and Durability, providing reliable transaction processing and data integrity.
- **Backup:** Azure SQL Database automatically manages backups, enabling point-in-time restore and long-term retention, ensuring data recovery in case of failure.

Encryption:

- **In Transit:** Uses Transport Layer Security (TLS) to encrypt data as it moves between the application and the database.
- **At Rest:** Employs Transparent Data Encryption (TDE) to encrypt data stored in the database, protecting sensitive information from unauthorized access.

Design Principles Used and Why

Single Responsibility Principle (SRP):

Each module is responsible for a single part of the functionality (e.g., authentication, expense submission).

Reason: Simplifies maintenance and enhances clarity and modularity.

Open/Closed Principle (OCP):

Modules are designed to be extendable without modifying existing code.

Reason: Facilitates the addition of new features with minimal risk of introducing bugs.

Dependency Inversion Principle (DIP):

High-level modules do not depend on low-level modules; both depend on abstractions.

Reason: Promotes loose coupling and enhances testability.

Interface Segregation Principle (ISP):

Interfaces are designed to be specific to the client's needs.

Reason: Prevents clients from being forced to implement methods they do not use.

Separation of Concerns (SoC):

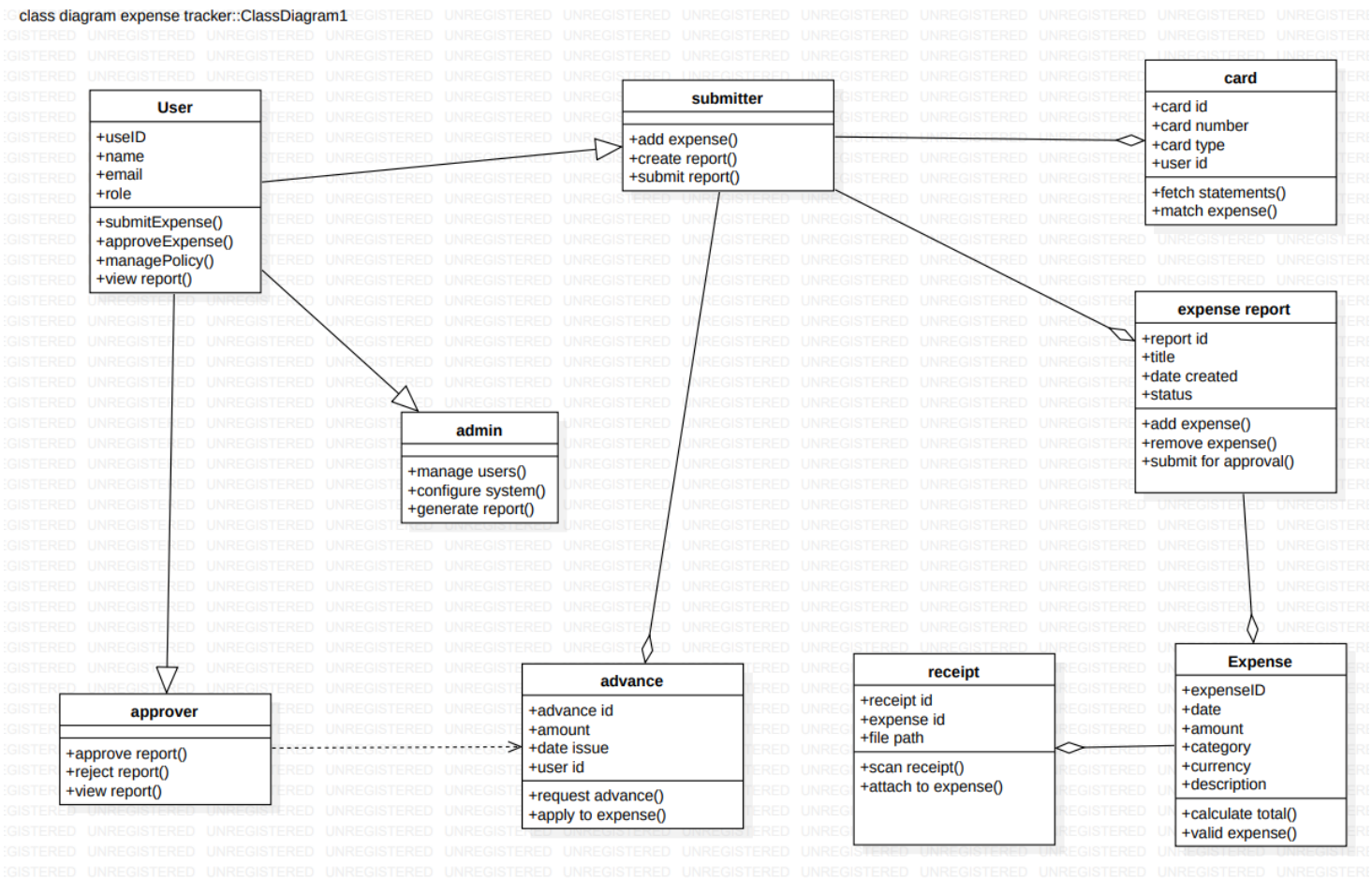
Different concerns (e.g., user interface, business logic, data access) are separated into distinct sections.

Reason: Enhances modularity, making the system easier to understand and manage.

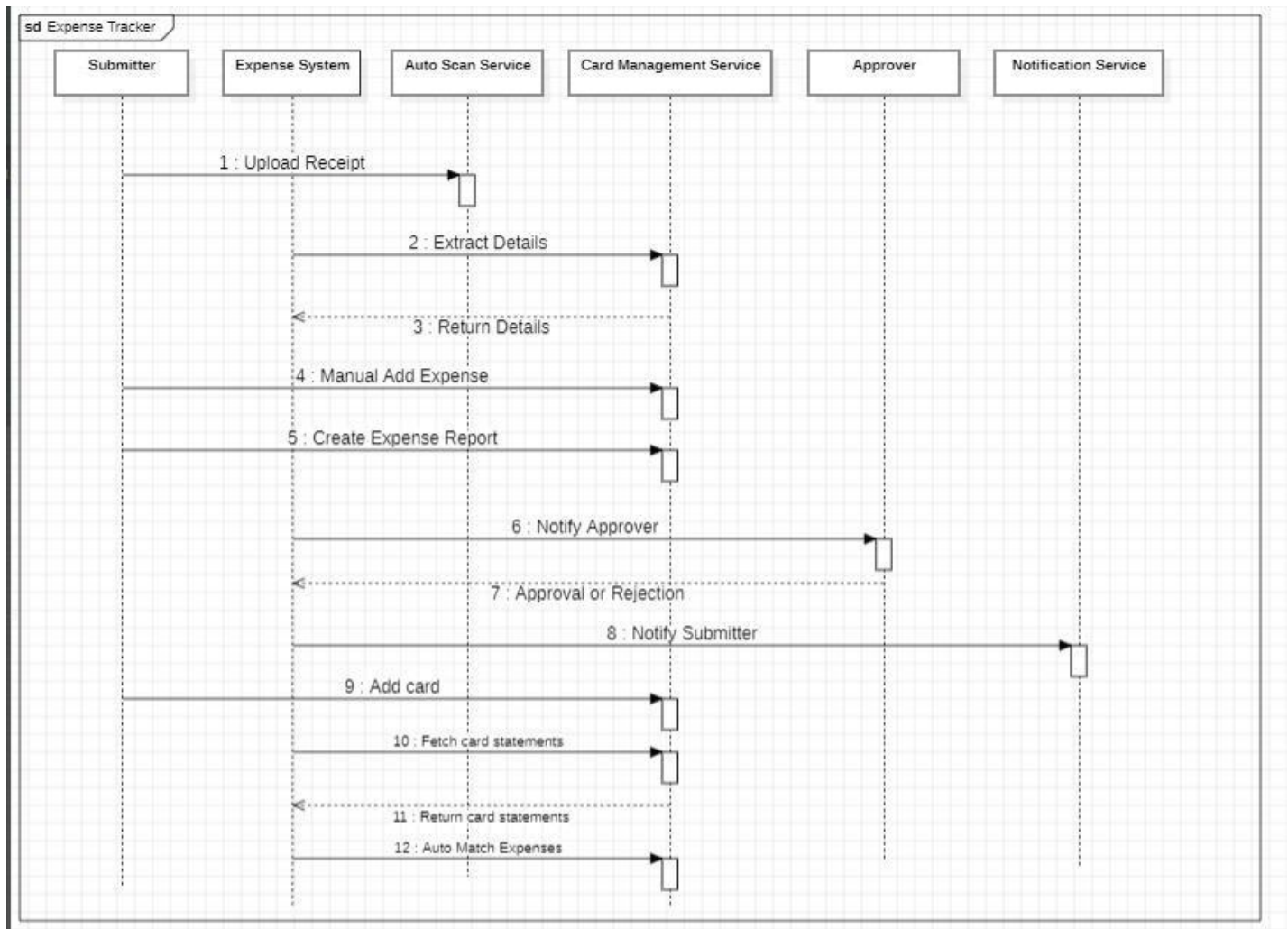
4.2 Architecture pattern used and why

The described architecture is a Microservices Architecture that supports ServiceOriented Architecture principles. It is designed to be highly scalable, resilient, and maintainable. This architecture fits well with modern CI/CD pipelines, allowing for continuous integration, automated testing, and continuous deployment, ensuring rapid and reliable delivery of new features and updates.

4.3 Class diagram



4.4 Sequence diagram



Ch 5. TEST STRATEGY

5.1 TEST PLAN

1. Introduction

This test plan outlines the strategy, scope, objectives, resources, and schedule for testing the JARVIS TechExpense application. The goal is to ensure the system functions as expected, is free from defects, and meets user requirements.

2. Scope

The scope of testing includes all major functionalities of the JARVIS TechExpense application, including user authentication, expense submission, status tracking, expense review, reporting, budget management, notifications, and data security.

3. Objectives

- Validate that the application meets all specified requirements.
- Ensure all functionalities work correctly and efficiently.
- Identify and resolve defects.
- Verify data security and integrity.
- Confirm compatibility across different devices and browsers.

4. Test Items

- User Authentication
- Expense Submission
- Expense Status Tracking
- Expense Review and Approval
- Reporting
- Budget Management
- Notifications
- Data Security

5. Test Types

- Functional Testing
- Integration Testing
- System Testing
- Regression Testing
- Performance Testing
- Security Testing
- Usability Testing

6. Test Environment

- Devices: Various desktops, laptops, tablets, and smartphones.
- Operating Systems: Windows, macOS, Linux, iOS, Android.
- Browsers: Chrome, Firefox, Safari, Edge.

- Database: Azure SQL Database.

7. Test Data

- User Accounts: Different types of user accounts (submitters, approvers) with varying levels of access and permissions.
- Expenses: Various expense entries with different categories, amounts, and statuses.
- Budget Limits: Different budget settings for various categories.
- Notifications: Test scenarios for different types of notifications.

8. Test Cases

1. **User Authentication** ◦ Test login with valid and invalid credentials. ◦ Test secure password handling (encryption, storage).
 - Test account lockout after multiple failed attempts.
2. **Expense Submission** ◦ Test submitting an expense with all required fields. ◦ Test validation errors for missing or incorrect data.
 - Test submission of expenses in different categories.
3. **Expense Status Tracking** ◦ Test tracking the status of submitted expenses. ◦ Test real-time updates on expense status changes.
4. **Expense Review and Approval** ◦ Test reviewing submitted expenses. ◦ Test approving and rejecting expenses.
 - Verify changes in expense status after approval/rejection.
5. **Reporting** ◦ Test generating reports for different periods. ◦ Test accuracy of the data in the reports.
 - Verify filtering and customization options in reports.
6. **Budget Management**
 - Test setting budget limits for different categories.
 - Test notifications for approaching/exceeding budget limits.
7. **Notifications** ◦ Test sending notifications for expense approvals/rejections.
 - Verify receipt of notifications by users.
8. **Data Security** ◦ Test data encryption in transit and at rest.
 - Test secure data access controls.
 - Test backup and restore procedures.

5.2 TEST CASE

1. User Authentication

Test Case 1.1: Valid Login

- **Objective:** Verify that users can log in with valid credentials.
- **Preconditions:** User account exists.
- **Steps:**
 1. Navigate to the login page.
 2. Enter valid username and password.
 3. Click on the "Login" button.
- **Expected Result:** User is successfully logged in and redirected to the dashboard.

Test Case 1.2: Invalid Login

- **Objective:** Verify that users cannot log in with invalid credentials.
- **Preconditions:** User account exists.
- **Steps:**
 1. Navigate to the login page.
 2. Enter invalid username and/or password.
 3. Click on the "Login" button.
- **Expected Result:** User is not logged in, and an error message is displayed.

Test Case 1.3: Account Lockout After Multiple Failed Attempts

- **Objective:** Verify that the account is locked after multiple failed login attempts.
- **Preconditions:** User account exists.
- **Steps:**
 1. Navigate to the login page.
 2. Enter invalid credentials multiple times (e.g., 5 times).
 3. Click on the "Login" button each time.
- **Expected Result:** Account is locked, and a lockout message is displayed.

2. Expense Submission

Test Case 2.1: Submit Expense with Valid Data

- **Objective:** Verify that users can submit an expense with valid data.
- **Preconditions:** User is logged in.
- **Steps:**
 1. Navigate to the expense submission page.
 2. Enter valid expense details (date, amount, description, category).
 3. Click on the "Submit" button.
- **Expected Result:** Expense is successfully submitted, and a confirmation message is displayed.

Test Case 2.2: Submit Expense with Missing Data

- **Objective:** Verify that the system prevents submission with missing required data.
- **Preconditions:** User is logged in.
- **Steps:**
 1. Navigate to the expense submission page.
 2. Leave one or more required fields blank (e.g., amount).
 3. Click on the "Submit" button.
- **Expected Result:** Expense is not submitted, and an error message indicating the missing fields is displayed.

3. Expense Status Tracking

Test Case 3.1: Track Expense Status

- **Objective:** Verify that users can track the status of their submitted expenses.
- **Preconditions:** User is logged in, and at least one expense has been submitted.
- **Steps:**
 1. Navigate to the expense tracking page.
 2. Locate a submitted expense.
- **Expected Result:** The current status (e.g., Pending, Approved, Rejected) of the expense is displayed.

4. Expense Review and Approval

Test Case 4.1: Approve Expense

- **Objective:** Verify that approvers can approve a submitted expense.
- **Preconditions:** User is logged in as an approver, and at least one expense is pending approval.
- **Steps:**
 1. Navigate to the expense review page.
 2. Select an expense and click on the "Approve" button.
- **Expected Result:** Expense status is updated to "Approved", and a confirmation message is displayed.

Test Case 4.2: Reject Expense

- **Objective:** Verify that approvers can reject a submitted expense.
- **Preconditions:** User is logged in as an approver, and at least one expense is pending approval.
- **Steps:**
 1. Navigate to the expense review page.
 2. Select an expense and click on the "Reject" button.
- **Expected Result:** Expense status is updated to "Rejected", and a confirmation message is displayed.

5. Reporting

Test Case 5.1: Generate Expense Report

- **Objective:** Verify that users can generate an expense report for a specified period.

- **Preconditions:** User is logged in.
- **Steps:**
 1. Navigate to the reporting page.
 2. Select a date range and any other filters.
 3. Click on the "Generate Report" button.
- **Expected Result:** Report is generated, displaying expenses for the selected period.

6. Budget Management

Test Case 6.1: Set Budget Limit

- **Objective:** Verify that users can set budget limits for expense categories.
- **Preconditions:** User is logged in.
- **Steps:**
 1. Navigate to the budget management page.
 2. Select a category and enter a budget limit.
 3. Click on the "Set Budget" button.
- **Expected Result:** Budget limit is set, and a confirmation message is displayed.

7. Notifications

Test Case 7.1: Receive Notification for Approved Expense

- **Objective:** Verify that users receive notifications when their expense is approved.
- **Preconditions:** User is logged in, and an expense has been approved by an approver.
- **Steps:**
 1. Wait for the approval notification.
- **Expected Result:** Notification is received indicating the expense approval.

8. Data Security

Test Case 8.1: Data Encryption in Transit

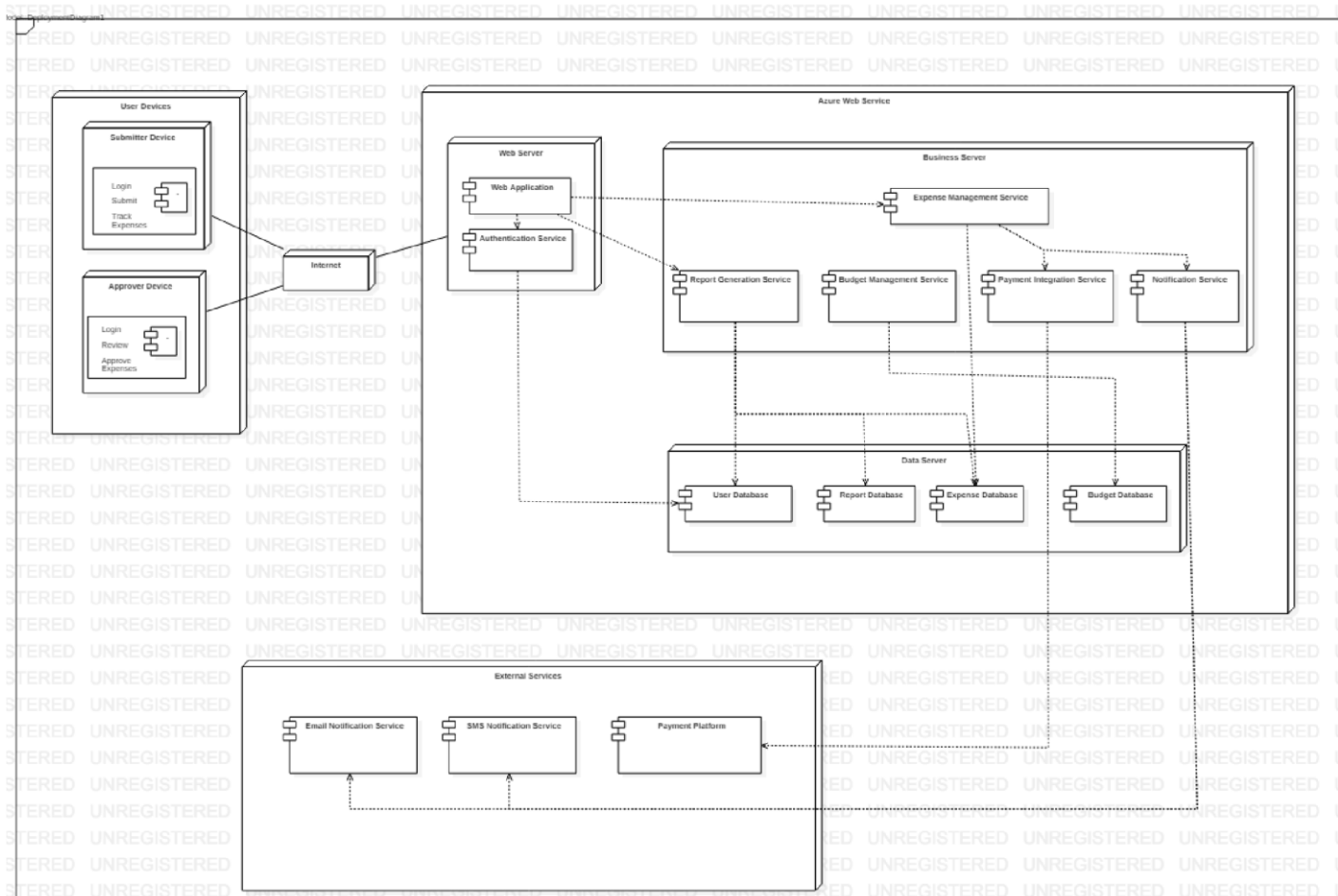
- **Objective:** Verify that data is encrypted during transmission.
- **Preconditions:** User is performing any data transmission action (e.g., logging in, submitting an expense).
- **Steps:**
 1. Use network monitoring tools to check the data transmission.
- **Expected Result:** Data is encrypted using TLS during transmission.

Test Case 8.2: Data Encryption at Rest

- **Objective:** Verify that data is encrypted at rest.
- **Preconditions:** Sensitive data is stored in the database.
- **Steps:**
 1. Inspect the database storage using appropriate tools.
- **Expected Result:** Data is encrypted using Transparent Data Encryption (TDE).

Ch 6. DEPLOYMENT

6.1 DEVELOPMENT ARCHITECTURE



6.2 AZURE SERVICES USED

- **Azure SQL Database:**
 - **Purpose:** Primary data storage for all structured data, including user information, expenses, reports, and logs.
 - **Features:** Provides ACID compliance, high availability, automatic backups, and encryption.
- **Azure Active Directory (Azure AD):**
 - **Purpose:** Authentication and user management.
 - **Features:** Secure login, multi-factor authentication, single sign-on, and user access management.
- **Azure App Service:**
 - **Purpose:** Hosting the web application and APIs.
 - **Features:** Auto-scaling, high availability, integrated security, and continuous deployment capabilities.
- **Azure Key Vault:**
 - **Purpose:** Storing and managing sensitive information such as API keys, passwords, and certificates.
 - **Features:** Provides secure key management and secrets management with access policies.
- **Azure Storage:**
 - **Purpose:** Storing unstructured data such as expense receipts, logs, and backups.
 - **Features:** Scalable and secure storage options with redundancy and encryption.
- **Azure Monitor:**
 - **Purpose:** Monitoring the performance and health of the application.
 - **Features:** Provides real-time insights, alerts, and log analytics.

- **Azure Logic Apps:**
 - **Purpose:** Automating workflows, such as sending notifications for expense approvals or rejections.
 - **Features:** Connects to various services and APIs, enabling automated workflows and integrations.
- **Azure Backup:**
 - **Purpose:** Ensuring data backup and disaster recovery.
 - **Features:** Provides backup and restore capabilities for Azure SQL Database and other critical data.
- **Azure Security Center:**
 - **Purpose:** Enhancing the security posture of the application.
 - **Features:** Continuous security assessment, threat protection, and compliance management.
- **Azure DevOps:**
 - **Purpose:** Managing the development lifecycle, including planning, coding, building, testing, and deployment.
 - **Features:** Provides CI/CD pipelines, version control, and project management tools.