Azure migration costs calculator with embedded AI

Software Requirements Specification (SRS)

CS 4850 - Section 02 – Fall 2024

Aug 21, 2024

Roles	Name	Major responsibilities	Contact
Project owner	Capstone		
Team leader	Kunal Shenoi	Organize meetings, monitor progress, and submit deliverables. Develop and implement testing protocols.	678-779-4770
Team members	Graham Allen	Developer - Developing back-end AI integration, implementing Azure/AWS API calls, and database connections. Assisting in front-end UI/UX development	770-841-6718
	Angel Hernandez	Developer - Designing and implementing a user-friendly interface, while assisting with backend and AI integration and development	770-318-5359
	Yvan Ngah	Documentation - Creating, maintaining, and ensuring the accuracy of all technical documentation.	171haden@gmail.com
Advisor / Instructor	Sharon Perry	Facilitate project progress; advise on project planning and management.	770-329-3895

Kunal Shenoi
Team Leader, Test

Graham Allen

Developer

Angel Hernandez
Developer

Yvan Ngah
Documentation

Table of Contents

Introduction

1.1 Overview

Our project focuses on developing an Azure Migration Cost Calculator with Embedded AI, integrated into a web application designed to accurately assess the financial impact of migrating on-premises infrastructure to Azure cloud services. The AI enhances traditional tools by offering precise cost predictions and customized recommendations, with reports generated based on each client's unique infrastructure and Azure pricing. The system features a chatbot interface for seamless user interaction, keeping insights up-to-date through continuous learning. The project aims to deliver a more efficient and informed migration process, targeting stakeholders such as IT teams, financial planners, and business decision-makers.

1.2 Project Goals

- AI-powered cost prediction and recommendations
- Integration with Azure Pricing APIs
- Detailed reports generation based on client infrastructure
- Chatbot interface for natural language interactions
- Continuous learning and update capabilities through AI

1.3 Definitions and Acronyms

1. Cloud Provider: A company that offers cloud computing services (e.g., AWS, Azure, Google Cloud).
2. Iaas (Infrastructure as a service): Cloud services that provide virtualized computing resources over the internet.
3. VM (Virtual Machine): A virtualized computing environment that behaves like a separate computer system.
4. IaC (Infrastructure as Code): The management of infrastructure (networks, virtual machines, load balancers, etc.) in a descriptive model, using the same versioning as DevOps team uses for source code.
5. Azure Pricing APIs: APIs provided by Microsoft Azure that offer real-time pricing information for various Azure services.
6. Migration: Moving data, applications, or other business elements from an on-premises infrastructure to a cloud-based one.
7. Continuous Learning: Constantly updating AI models with new data to improve their predictions and recommendations.
8. CI/CD (Continuous Integration/Continuous Deployment): Practices that involve automatically testing, integrating, and deploying code changes to reduce errors and accelerate development.

1.4 Assumptions

- The users will be familiar with basic cloud computing concepts.
- Azure pricing APIs will be publicly available.
- The application will be web-based, and accessible via modern browsers.

Design Constraints

2.1 Environment

- Development Tools: The project will utilize industry-standard tools and frameworks, including React or Angular for the front end and Flask for the back end, to ensure robust and scalable development. By leveraging open-source technologies, we aim to enhance flexibility, reduce costs, and foster collaboration, while ensuring the application is built on a solid, well-supported foundation.
- Operating System Compatibility: The web application must be accessible across multiple operating systems (Windows, macOS, Linux) through modern web browsers (Chrome, Firefox, MS Edge, Safari)

2.2 User Characteristics

- Technical proficiency: Users of this application will have varying levels of technical expertise. Some users may possess in-depth knowledge of cloud computing concepts while others may be analysts or stakeholders with limited technical knowledge. The design must cater to a diverse group of users by offering a user-friendly interface that simplifies complex calculations while still providing advanced features for expert users.
- Accessibility requirement: The application must adhere to accessibility standards to ensure it is usable by users with disabilities. This includes considerations such as keyboard navigation, color contrast, text-to-speech, and scalable text.

2.3 System

- Scalability: The system must be designed to handle varying loads, from small-scale migrations involving a few servers to large enterprise-level migrations. This constraint requires that the architecture supports horizontal scaling, allowing the application to add or remove resources based on demand.
- Performance: The application must deliver migration cost calculations and AI-driven recommendations in real-time or near-real-time. Considering the large performance requirements of calculations and migrations the use of HPC services or optimized algorithms will be required for optimal performance.
- Integration with External Services: The application must integrate seamlessly with Azure Pricing APIs, other third-party APIs, and potentially other cloud computing services.
- Security and Compliance: Given the sensitive nature of the data, the application must adhere to strict security protocols. This includes data encryption, secure authentication methods, and compliance with industry security standards.

Functional Requirements

3.1 User Registration and Authentication

- 3.1.1 Account:
    - The system must provide user registration functionality where users can create an account by providing necessary details such as a username and password.

    - User authentication must be handled securely, ensuring that passwords are encrypted using industry standards.
    - The system must support multi-factor authentication(MFA), providing an additional layer of security.
    - User roles must be defined in addition to permissions and access levels for users, administrators, and other roles.
    - The system must store or cache user sessions ensuring quick restorations to sessions. In addition, sessions should expire after a period of inactivity by users.
    - The system should provide simple but effective password management for users and accounts.

3.2 Report Generation and Export

- 3.2.1 Detailed Cost Reports:
    - The system must generate comprehensive reports that detail the estimated migration costs, broken down by resource type, service, and region.
    - The reports should include charts and graphs, to help users easily interpret the data.
    - Users should be able to customize the report contents by selecting specific data points or sections to include.
- 3.2.2 Export Options:
    - The reports should be exportable in multiple formats, including PDF, Excel, and CSV.

3.3 User Dashboard

- 3.2.1 Personalized Dashboard:
    - Upon logging in, users should be greeted with a personalized dashboard that summarizes key information such as recent cost assessments, saved reports, and upcoming milestones.
    - The dashboard should display alerts for any updates in Azure pricing or significant changes in the user's infrastructure that could impact migration costs.
    - The dashboard should allow simple switching between accounts within the organization.
- 3.2.2 Activity Log:
    - The system should maintain a detailed log of all user activities, such as data inputs, scenario runs, and report generation.

3.4 Notifications and Alerts

- 3.4.1 Pricing Updates:
    - The system should send notifications to users when there are significant updates to Azure pricing that may impact their migration costs.
- 3.4.2 Infrastructure Changes:
    - The system must notify users of significant changes in their on-premises infrastructure that could impact migration costs.