Empowering Event Networking and Social Interaction: A Cloud Mobile App-Meet You

Group members：Xiaoyu Zhou (Ann) 1565763, Bo Huang (Julian) 1543542, Jiqiang Wang (Robert)1564710

**Introduction:**

In today's fast-paced world, attending events presents a unique opportunity for individuals to connect, share experiences, and foster new friendships. However, the challenges of limited time and the inability to easily identify like-minded individuals often hinder the potential for meaningful interactions. To overcome these obstacles, the development of a mobile application that leverages cloud computing capabilities offers a promising solution. This essay explores the computing problem and discusses how a cloud computing framework can address it. Additionally, a vision statement is formulated to encapsulate the app's overarching goal.

**Computing Problem:**

The computing problem at hand revolves around the need for an efficient and effective method to facilitate social interactions and networking among event participants. Traditional methods fall short in enabling individuals to identify and connect with others who share similar interests and attend the same events. Additionally, logistical constraints hinder real-time communication and inhibit the ability to share location information within the event.

**Solution with Cloud Computing Framework:**

Implementing a cloud computing framework presents a viable solution to address the computing problem outlined above. The following aspects highlight how cloud computing can effectively tackle the challenges:

Scalability: Cloud computing provides the ability to scale resources dynamically based on event demand. This ensures the app can handle fluctuating user loads, accommodating both small and large-scale events seamlessly.

Data Management: A robust backend powered by cloud-based storage services and databases can efficiently manage user profiles, event details, and location data. Cloud infrastructure offers scalability, high availability, and reliable data storage, ensuring optimal performance and seamless user experiences.

Real-Time Communication: Cloud-based communication tools, such as WebSockets or push notifications, enable instant and real-time communication among event participants. This facilitates connection-building, allowing users to interact, exchange messages, and share their experiences within the app.

Location Services: Cloud platforms provide location-based services that can be leveraged to track and share user locations within events. By integrating geolocation data, mapping functionalities, and proximity-based notifications, the app can enhance social interactions and facilitate in-person meetups.

Security and Privacy: Cloud providers prioritize security measures, including encryption, access controls, and regular monitoring. By leveraging these robust security features, user data can be protected, ensuring privacy and maintaining user trust.

**Vision Statement:**

"Our vision is to create a mobile app that revolutionizes event participation, empowering individuals to effortlessly connect, forge new friendships, and enhance their event experiences through seamless networking and location sharing. By harnessing the capabilities of cloud computing, our app aims to provide a scalable, secure, and real-time platform, facilitating meaningful interactions, efficient data management, and robust communication. Our goal is to foster a vibrant community of event-goers, where connections are easily formed, enriching the overall event experience with the power of technology."

Product backlog

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Name** | **User story** | **Tasks** | **Priority** |
| 1 | Register and login | As a user, I want to create an account and profile so that I can join events and connect with other attendees. | 1. Design and develop a user registration page with fields for username, email address, and password. 2. Implement server-side validation to ensure that the username, email address, and password meet the required criteria (e.g., minimum length, valid email format, strong password requirements). 3. Store the user registration data securely in a database. 4. Send a verification email to the user's provided email address to confirm their account. 5. Create a user login page with fields for username/email and password. | High |
| 2 | Search for events | As a user, I want to search for events happening in my area and view event details such as date, time, and location. | 1. Design and develop a search functionality where users can search for events based on location, keywords, or categories. 2. Implement a database query to fetch events based on the user's search parameters. 3. Display a list of search results showing relevant event details such as event name, date, time, location, and a brief description. 4. Create an event detail page that provides more information about a selected event, including additional details like event organizers, guest speakers, agenda, and any associated media (e.g., images or videos). | High |
| 3 | Purchase a ticket | As a user, I want to join an event by RSVPing or purchasing tickets within the app. | 1. Create a user-friendly interface to allow users to RSVP or purchase tickets for an event. 2. Implement a secure payment gateway integration to facilitate ticket purchases, if applicable. 3. Develop a system to generate unique ticket codes or confirmation numbers for users who successfully RSVP or purchase tickets. 4. Store the user's RSVP or ticket information in the database and associate it with their user profile and the specific event. | High |
| 4 | Participant list | As a user, I want to see a list of attendees for an event and have the ability to contact them through messaging or social media integration. | 1. Develop a feature to display a list of attendees for each event, either publicly or within the app for registered users. 2. Implement a user interface that allows users to view attendee profiles and access contact information, subject to privacy settings. 3. Integrate a messaging system that enables users to send messages to other attendees directly through the app or via email, if allowed by privacy settings. 4. Provide options for users to connect their social media accounts and share their profiles with other attendees, facilitating communication outside the app. | High |
| 5 | Sharing the location | As a user, I want to share my location with other attendees during the event to facilitate meetups and networking opportunities. | 1. Implement a feature that allows users to voluntarily share their location during an event. 2. Integrate a mapping service to display the location of users who have opted to share their location on a map within the app. 3. Provide privacy settings that allow users to control when and with whom their location information is shared. | Medium |
| 6 | Notification | As a user, I want to receive notifications about event updates, changes in schedule, or important announcements. | 1. Develop a notification system that sends relevant event updates and announcements to users. 2. Implement push notifications for mobile devices or email notifications for users who prefer to receive updates via email. 3. Provide options for users to customize their notification preferences, such as frequency or types of notifications. | Medium |
| 7 | Feedback | As a user, I want to provide feedback and ratings for events and share my experience with other potential attendees. | 1. Create a user interface that allows users to provide feedback and ratings for events they have attended. 2. Store user feedback and ratings in the database and associate them with the respective events. 3. Develop a feature that allows users to view and read feedback and ratings provided by other attendees for a particular event. 4. Implement a system to filter and moderate user feedback to ensure relevance and appropriateness. | Low |

**User interfaces**

A screen shot of a cell phone

Description automatically generated with medium confidence A screenshot of a login screen

Description automatically generated with medium confidence A screenshot of a login screen

Description automatically generated with medium confidence A screenshot of a phone

Description automatically generated with medium confidence

A screenshot of a phone

Description automatically generated with medium confidence A screenshot of a phone

Description automatically generated with medium confidence A screenshot of a phone

Description automatically generated with medium confidence

A screenshot of a phone

Description automatically generated with medium confidence A screenshot of a phone

Description automatically generated with low confidence A screenshot of a phone

Description automatically generated with medium confidence

A screen shot of a phone

Description automatically generated with medium confidence A screenshot of a phone

Description automatically generated with medium confidence A screenshot of a phone

Description automatically generated with medium confidence

A screenshot of a phone

Description automatically generated with medium confidence

**Cloud Service List:**

|  |  |
| --- | --- |
| **Service** | **Reason** |
| Azure Active Directory (Azure AD) | Azure AD provides robust user authentication and authorization services, allowing secure user registration, login, and authentication functionality for the application. It ensures that only authorized users can access the application and their respective profiles. |
| Azure Cosmos DB-SQL | Azure Cosmos DB is a globally distributed, multi-model database service that can be used as the data store to persist all data. It offers scalability, low-latency access, and supports various data models (such as SQL, NoSQL, graph, etc.) to efficiently store and retrieve user registration data, event information, attendee details, and feedback. |
| Azure Storage | Azure Storage can be used as a cloud storage service to store and serve user profile pictures and event media (e.g., images, videos). It provides scalable and durable storage options, such as Blob Storage, for efficient management of media assets within the application. |
| Azure Functions | Azure Functions can be utilized to implement serverless functions that handle specific application logic or event triggers. For example, functions can be used to process ticket purchases, send notifications, or perform other backend operations efficiently. |
| Azure Notification Hubs | Azure Notification Hubs is a scalable push notification service that allows you to send notifications to users across multiple platforms (e.g., iOS, Android, Windows) and devices. It can be used to deliver event updates, changes in schedule, or important announcements to users in real-time. |
| Azure Maps | Azure Maps provides geolocation and mapping services that can be integrated into the application. It allows users to search for events in their area, share their location during events, and facilitates meetups and networking opportunities by providing mapping functionality for event venues and attendee locations. |
| Azure SignalR Service | Azure SignalR Service enables real-time messaging and chat functionality within the application. It can be used to implement the ability for attendees to contact each other directly through the app, fostering communication and networking opportunities during events. |
| Azure Event Grid | Azure Event Grid can be utilized to handle event-driven scenarios within the application. It can help in capturing and processing events related to ticket purchases, user RSVPs, or other important actions, enabling efficient event-driven communication and integration across various components of the system. |
| Azure Logic Apps | Azure Logic Apps provide a visual development environment for building workflows and integrating various services and APIs. It can be used to automate tasks such as sending emails, processing user feedback, or triggering actions based on specific events or conditions. |
| Azure SendGrid | Azure SendGrid, an email delivery service, can be used to send verification emails during user registration and notify users about important account-related actions. It provides a reliable and scalable solution for sending transactional emails within the application. |

**Architectural Overview:**

The application architecture will be based on a cloud-native approach using various Azure services to build a scalable and robust event management and networking platform. Here is an overview of the architecture:

User Interface (UI):

The UI will be implemented using front-end technologies like HTML, CSS, and JavaScript frameworks.

It will provide an intuitive user interface for users to interact with the application, including features such as user registration, event search, attendee listing, messaging, and profile management.

Azure Active Directory (Azure AD):

Azure AD will be used for user authentication and authorization.

It provides secure user registration, login, and authentication services, ensuring that only authorized users can access the application and their respective profiles (Chappell, 2008).

Azure Cosmos DB:

Azure Cosmos DB will serve as the primary data store for the application, providing a globally distributed, highly scalable, and multi-model database service.

User registration data, event information, attendee details, and feedback will be stored and retrieved from Azure Cosmos DB.

It offers different data models (e.g., SQL, NoSQL, graph) to efficiently handle diverse data requirements.

Azure Storage:

Azure Storage will be utilized as a cloud storage service to store and serve media assets, including user profile pictures and event media (e.g., images, videos) (Azure, 2016).

Blob Storage within Azure Storage provides scalable and durable storage options for managing media assets efficiently.

Azure Functions:

Azure Functions will be used for implementing serverless functions that handle specific application logic or event triggers.

Functions can be employed to process ticket purchases, send notifications, or perform backend operations efficiently.

Azure Notification Hubs:

Azure Notification Hubs will be used to send push notifications to users across multiple platforms (iOS, Android, Windows) and devices(Azure, 2016).

It enables the delivery of event updates, changes in schedule, or important announcements to users in real-time.

Azure Maps:

Azure Maps will be integrated to provide geolocation and mapping services within the application.

It enables users to search for events in their area, share their location during events, and facilitates meetups and networking opportunities.

Azure SignalR Service:

Azure SignalR Service will be used to enable real-time messaging and chat functionality.

It allows attendees to contact each other directly through the app, fostering communication and networking during events.

External APIs:

The application will connect to external APIs for additional functionality, such as social media integration (e.g., Facebook Login, Twitter API) to allow users to import relevant information or contacts to their profiles and facilitate communication outside the app.

Payment gateway integration APIs (e.g., Stripe, PayPal) will be utilized to handle secure ticket purchases and payment processing.

APIs from mapping services like Google Maps API or Mapbox can be integrated for enhanced geolocation and mapping features.

**Data Communication:**

The UI will communicate with Azure AD for user authentication and authorization.

The UI will interact with Azure Cosmos DB to retrieve and store user data, event information, attendee details, and feedback.

Azure Functions will communicate with Azure Cosmos DB and other services to handle specific application logic and event triggers.

Azure Notification Hubs will send push notifications to users, triggered by events from Azure Cosmos DB or other relevant services.

Azure Maps will be utilized for geolocation services, providing event search functionalities and facilitating meetups.

Azure SignalR Service will enable real-time messaging and chat functionalities between attendees.

External APIs will be integrated into the application to connect with social media platforms, payment gateways, and mapping services.

**Project Plan:**

1. Define Project Scope and Objectives:

Clearly define the scope of the project, including the target audience, key features, and overall objectives.

Identify the desired outcomes and success criteria for the application.

Target Audience:

The mobile application targets individuals attending various events, conferences, and gatherings who are seeking to connect with like-minded individuals, share experiences, and build new friendships. The app aims to cater to a diverse range of interests, including professional, social, educational, and recreational events.

Key Features:

User Profiles: Users can create profiles with relevant information such as their interests, professional background, and personal preferences to facilitate meaningful connections.

Event Discovery: The app will provide a comprehensive database of events across different categories, allowing users to explore and find events that align with their interests and schedule.

Matchmaking Algorithm: A sophisticated matchmaking algorithm will analyze user profiles and event details to suggest potential connections based on shared interests, goals, and location within the event.

Real-time Messaging: The app will enable users to communicate and initiate conversations with their matched connections in real-time, facilitating easy and convenient networking during events.

Personalized Recommendations: Based on user preferences and previous interactions, the app will provide personalized event recommendations and suggest relevant connections for future events.

Event Feedback and Ratings: Users will have the ability to provide feedback and rate events, speakers, and overall experiences, helping others make informed decisions and fostering a community-driven feedback system.

Overall Objectives:

Enhance Networking Opportunities: The primary objective of the app is to empower individuals attending events to connect with like-minded individuals, fostering new friendships, professional relationships, and collaboration opportunities.

Improve Event Experience: By providing personalized event recommendations, facilitating relevant connections, and enabling real-time communication, the app aims to enhance the overall event experience for users.

Increase Engagement and Retention: The app intends to promote active user engagement and increase user retention by offering valuable networking features, personalized recommendations, and a user-friendly interface.

Expand User Base and Event Coverage: The project aims to attract a diverse user base and establish partnerships with event organizers to ensure a wide range of event coverage across different locations and interests.

1. Conduct Requirements Gathering:

Engage with stakeholders to gather detailed requirements, including user stories, functional and non-functional requirements, and any specific design preferences.

Prioritize requirements based on their importance and impact on the application's core functionality.

1. Design Application Architecture:

Design the application architecture based on the architectural overview, selecting the appropriate Azure services and external APIs.

Determine the data models, data flow, and communication patterns between different components.

Create a system architecture diagram to illustrate the overall structure of the application.

1. Develop User Interface (UI):

Start developing the user interface based on the finalized design and requirements.

Implement responsive and intuitive UI using front-end technologies like HTML, CSS, and JavaScript frameworks.

Continuously test and refine the UI based on user feedback and usability testing.

1. Implement Back-End Functionality:

Develop server-side functionality, including user authentication, event management, attendee listing, messaging, and profile management.

Utilize appropriate Azure services (Azure AD, Azure Cosmos DB, Azure Storage, Azure Functions, etc.) and external APIs for backend integration.

Ensure the implementation follows best practices for security, scalability, and performance.

1. Integrate External APIs:

Integrate external APIs such as social media APIs for user profile integration and communication, payment gateway APIs for ticket purchases, and mapping APIs for geolocation and mapping services.

Configure API credentials, implement API endpoints, and handle authentication and data exchange with external systems.

1. Test and Quality Assurance:

Develop and execute a comprehensive testing strategy, including unit testing, integration testing, and end-to-end testing.

Perform user acceptance testing (UAT) to ensure the application meets the desired functionality and user experience requirements.

Conduct performance testing to evaluate system scalability and responsiveness.

1. Deployment and Hosting:

Prepare the application for deployment to a hosting environment such as Azure App Service or Azure Kubernetes Service (AKS).

Set up appropriate configuration, environment variables, and deployment scripts.

Deploy the application to the chosen hosting environment and verify its functionality in the production environment.

1. User Training and Documentation:

Create user guides and documentation to assist users in navigating and utilizing the application's features.

Conduct user training sessions or create video tutorials to familiarize users with the application's functionalities and workflows.

1. Launch and Monitoring:

Promote the application and make it available to users.

Monitor the application's performance, user feedback, and analytics to identify areas for improvement and optimization (Wilder, 2012).

Address any post-launch issues or bugs reported by users and continuously iterate based on user feedback.

1. Maintenance and Support:

Provide ongoing maintenance and support for the application, including bug fixes, feature enhancements, and security updates.

Monitor and manage cloud services, ensuring they are up to date and properly scaled to handle user demand.

Regularly review and refine the application based on user feedback and changing requirements.

**Conclusion:**

The development of a mobile app powered by cloud computing offers a transformative solution to the computing problem of enabling event networking and social interaction. By leveraging the scalability, real-time communication, location services, and security features provided by cloud computing frameworks, individuals can seamlessly connect, make friends, and share their experiences within the app. This app, driven by a vision to enhance event participation, has the potential to redefine how people engage with events and build lasting connections in the digital age.

**Appendix 1**

Wireframe design:

Login & Profile

 

Event List & Event detail

 

Event ticket purchase & Event attendance

 

Location share & notification



Feedback



**Appendix 2**

Figma Demo design

https://www.figma.com/proto/HbxwlmFBK54dbLrAlEmW2b/MeetYou-APP?type=design&node-id=113-969&scaling=scale-down&page-id=0%3A1&starting-point-node-id=113%3A969

**Reference**

Chappell, D. (2008). Introducing the Azure services platform. White paper, Oct, 1364(11).

Azure, M. (2016). Microsoft azure. línea]. Available: https://docs. microsoft. com/es-es/azure/virtual-machines/linux/quick-createportal.[Último acceso: 10 Diciembre 2017].

Wilder, B. (2012). Cloud architecture patterns: using microsoft azure. " O'Reilly Media, Inc.".