Near Synonyms: High Level Design

Team Information

Team: Synonym Masters

Members: Itay riche, Yarden Gal Tahor, Yarin Ben-Zimra, Shay Bar, Guy Binyamin

System's Main Components

User Interface (UI):

Built with React, allowing users to interact with the system.

Backend Server:

Node.js for handling requests and business logic.

Database:

AWS database for storing sentences, synonyms, and user responses.

NLP Models:

Trained in Python, responsible for determining the appropriateness of synonyms.

Main Users' Use Cases

Persona: Language Learner

1. Login: The user logs into the application.

2. Quiz Start: The user starts a synonym quiz.

3. Question Display: A sentence with a missing word and two synonyms are displayed.

4. Answer Selection: The user selects the synonym they believe is most appropriate.

5. Feedback: The application provides feedback on the choice.

6. Result Storage: The user's answer is stored in the database for further analysis.

Front End Technology

Technology: React Native

Architecture: Component-based, with a Node.js backend for server-side logic.

Mocks for Main Pages

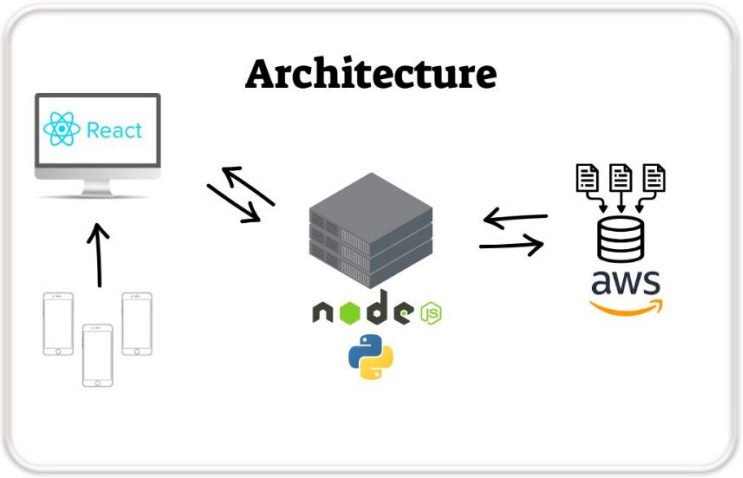
Login Page: Simple interface for user authentication.

Dashboard: Overview of the user's progress and options to start new quizzes.

Quiz Page: Displays sentences with missing words and synonym options.

Feedback Page: Shows correct answers and explanations.

System Architecture Overview



Component Breakdown

UI (React): Interactive components for user interaction. Communicates with the backend via RESTful APIs.

Backend (Node.js): Handles requests from the UI. Interfaces with the database and NLP models.

Database (AWS): Stores sentences, synonyms, and user responses. Ensures data integrity and availability.

NLP Models (Python): Trained on large datasets to evaluate synonym appropriateness. Integrated with the backend to provide real-time analysis.

Data Model and Storage

Data Schema:

Tables:

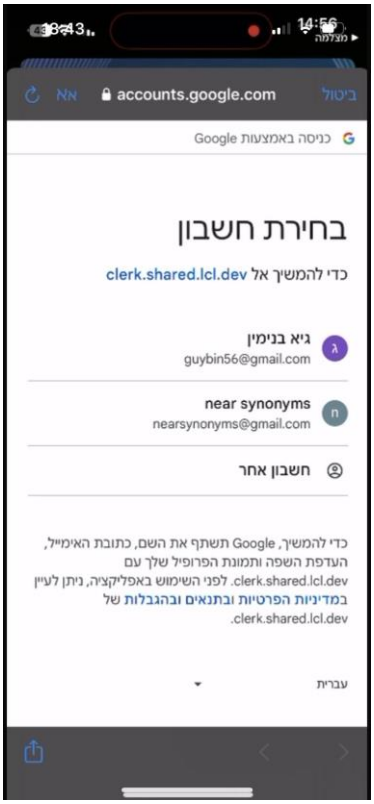
- Sentences: Stores sentence structures and missing words.
- Synonyms: Stores synonyms and their relationships.
- User Responses: Records user answers and timestamps.
- Storage: Hosted on AWS for scalability and reliability.

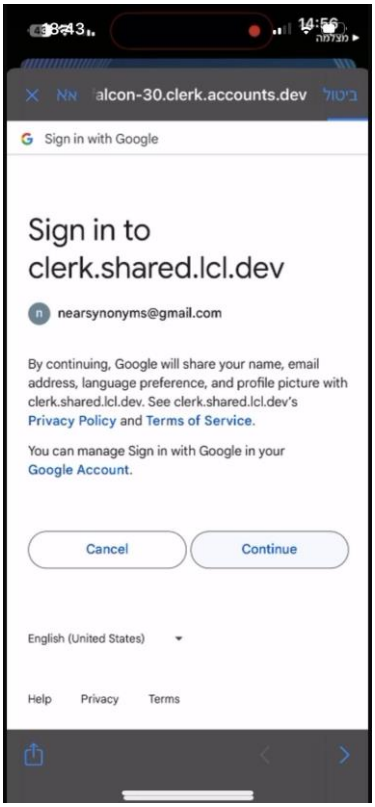
User Interface Design

Design Principles: User-friendly, intuitive, and responsive design.

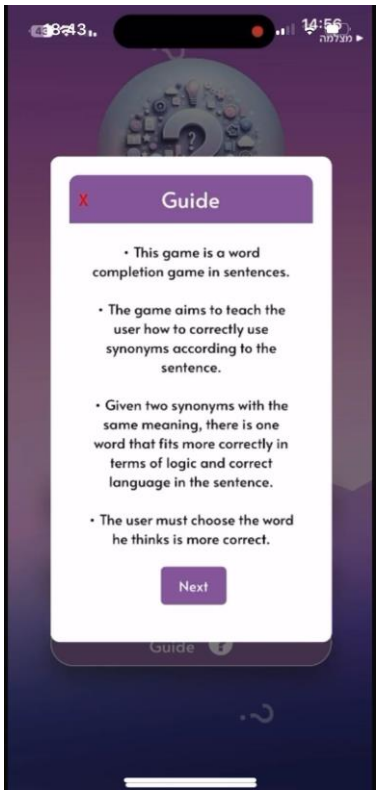
Consistent look and feel across all pages.

Mockups: High-level sketches of the main screens, focusing on usability.

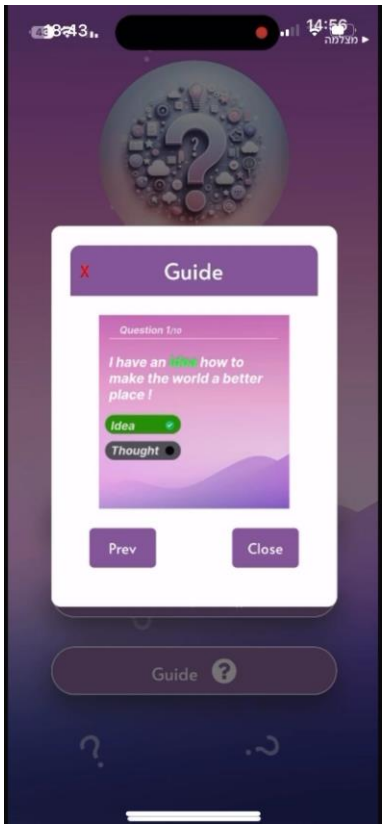




SYNONYMS
GAME
Play
Settings
Guide

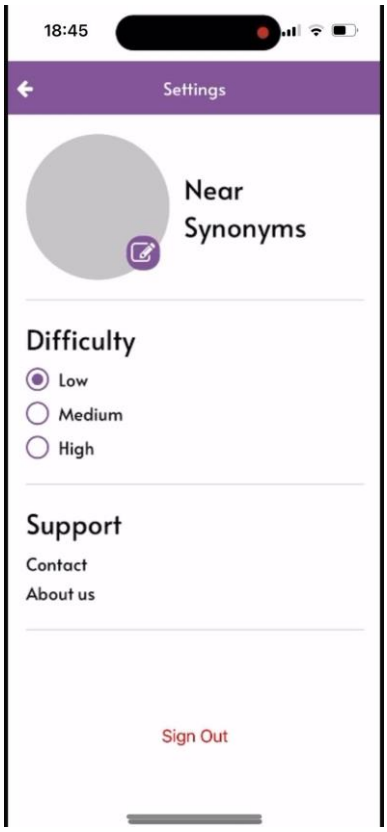

Guide
• This game is a word completion game in sentences.
• The game aims to teach the user how to correctly use synonyms according to the sentence.
• Given two synonyms with the same meaning, there is one word that fits more correctly in terms of logic and correct language in the sentence.
• The user must choose the word he thinks is more correct.
Next

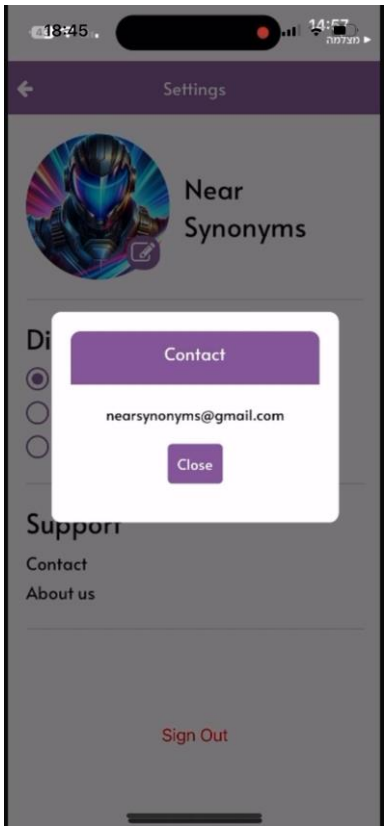
Guide
Question 1/10
I have an _____ how to make the world a better place !
Idea
Thought
Prev
Next
Guide

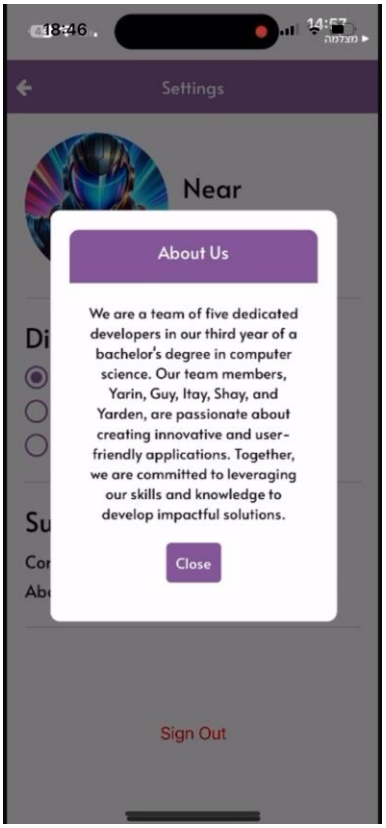

Guide
Question 1/10
I have an idea how to make the world a better place !
Idea
Thought
Prev
Close
Guide

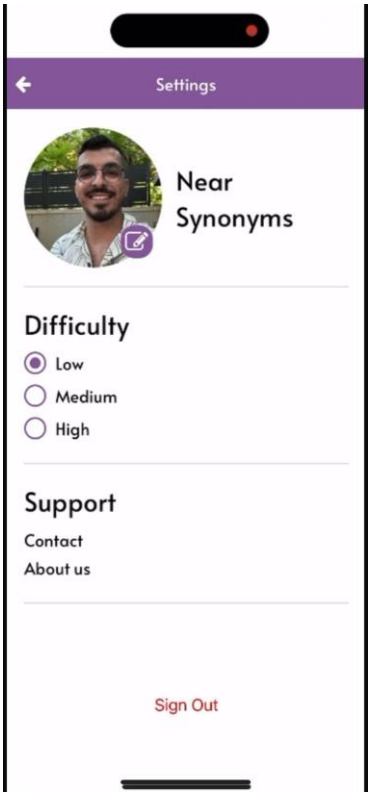
27
Question 1/10
I have an _____ how to make the world a better place !
Idea
Thought

27
Question 1/10
I have an idea how to make the world a better place !
Idea
Thought


18:45
Settings
Near
Synonyms
Difficulty
Low
Medium
High
Support
Contact
About us
Sign Out


18:45
Settings
Near
Synonyms
Contact
nearsynonyms@gmail.com
Close
Contact
About us
Sign Out


18:46
Settings
Near
About Us
We are a team of five dedicated developers in our third year of a bachelor's degree in computer science. Our team members, Yarin, Guy, Itay, Shay, and Yarden, are passionate about creating innovative and user-friendly applications. Together, we are committed to leveraging our skills and knowledge to develop impactful solutions.
Close
Sign Out


Settings
Near
Synonyms
Difficulty
Low
Medium
High
Support
Contact
About us
Sign Out

Integration Strategy

Approach: RESTful APIs for communication between frontend and backend.
Secure connections to the AWS database.

Protocols: HTTPS for secure data transmission. JSON format for data exchange.

Security Measures

Protocols: SSL/TLS for secure data transmission.
User authentication and authorization mechanisms.

Data Privacy: Data encryption at rest and in transit. Regular security audits and updates.

Scalability and Performance

Considerations: Load balancing and auto-scaling on AWS.
Optimized database queries and indexing.

Strategies: Caching frequently accessed data. Asynchronous processing for heavy tasks.

Error Handling and Recovery

Plan: Centralized error logging and monitoring. Automated recovery scripts for common issues.

Procedures: Regular backups and data recovery drills.

Deployment Strategy

Environment: Hosted on AWS cloud.

Steps: Continuous Integration and Continuous Deployment (CI/CD) pipeline. Automated tests and deployment scripts.

Testing and Quality Assurance

Methodologies: Unit, integration, and end-to-end testing. Code reviews and static analysis.

Environments: Staging and production environments for thorough testing.

Maintenance and Support

Plan: Regular updates and patch management.

Procedures: Bug tracking and resolution. Feature enhancement based on user feedback.

Performance Metrics and Monitoring

Metrics: Response time, error rates, and user engagement.

Monitoring Tools: AWS CloudWatch, application performance monitoring (APM) tools.

Regulatory and Compliance Considerations

Compliance: GDPR, CCPA, and other relevant data protection laws.

Alignment: Regular compliance checks and documentation.

Dependencies and External Services

List: Third-party NLP libraries, React Native and Node.js packages, AWS services.