

1. The product backlog

- **Learning Module**
 - **Flash Card**: Display a Chinese idiom along with its pinyin, meaning, and an example sentence.
 - **Idiom Collection**: Display a comprehensive list of collected idioms of each user.
 - **Idiom Search**: Search idioms based english descriptions
- **Game Module**
 - **Real-time Battles**: Compete with a random player in a timed challenge. The two players will be given 10 questions and 20 seconds to answer each question. All questions are single choice. There are two types of questions. One is to choose the closest idiom according to the English description of the idiom, and the other is to choose the closest English interpretation according to the idiom. During the game, a player can see his own and the opponent's progress and scores. The player with the higher score wins.
 - **Self-testing**: Randomized questions for personal practice. The type of problem is the same as described earlier.

2. The first sprint backlog

List of the functionalities:

- Task 1: Design and implement the main page
- Task 2: Design and implement pages related to Chinese Idiom Learning, including flash cards, idiom search
- Task 3: Design and implement the static pages related to the Game Module, including the battle and self-testing pages
- Task 4: Database design and data model representation
- Task 5: Implement models.py

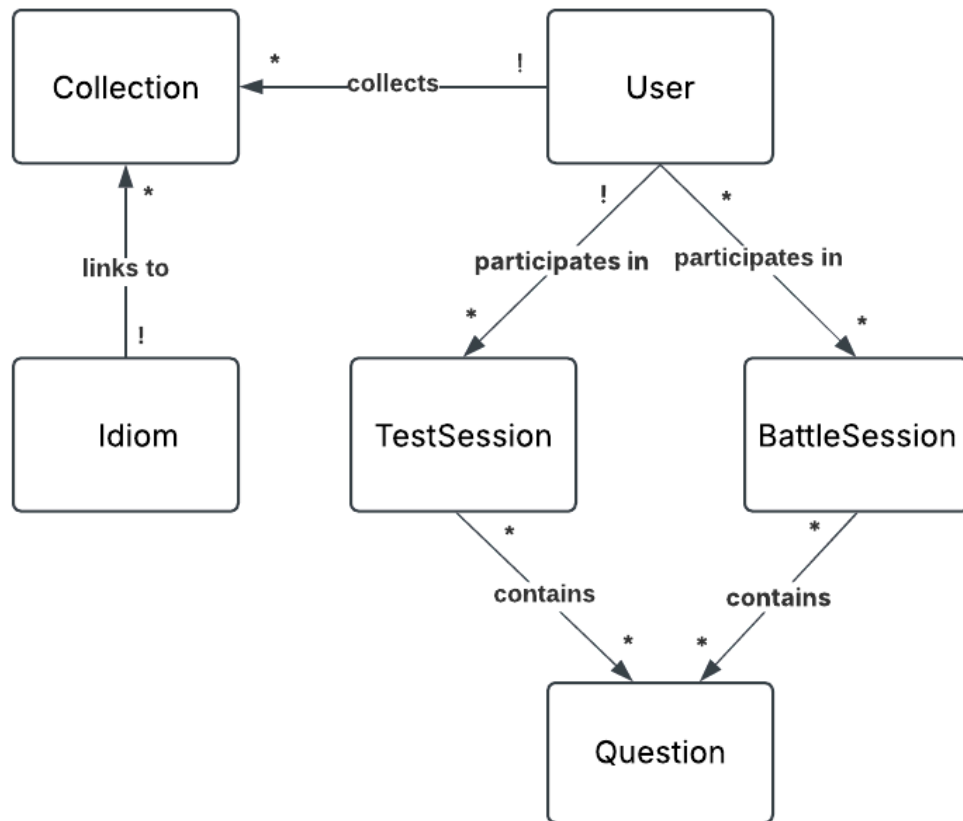
We have assigned two team members (Yuling Hu and Yi Cheng) to handle front-end development and two (Qianhui Long and Cynthia Yang) to focus on back-end development. The front-end team will be responsible for designing the application's mock-ups (Task 1-3), while the back-end team will manage the database design and model implementation (Task 4-5).

3. The name and Andrew ID of the product owner for the first sprint.

Yuling Hu (yulingh)

4. A complete specification of the data models used by your application.

Data Model:



models.py:

```
from django.db import models

class User(models.Model):
    """
    Stores user information.
    """
    username = models.CharField(max_length=50)
    password = models.CharField(max_length=255) # Hashed password

class Idiom(models.Model):
    """
    Stores Chinese Idiom.
    """
    idiom = models.CharField(max_length=100) # Idiom (Chinese)
    pronunciation = models.CharField(max_length=50) # Pinyin of the idiom
```

```

translation = models.CharField(max_length=100) # Direct translation of
the idiom
explanation = models.TextField() # Explanation of the idiom
updated_time = models.DateTimeField(auto_now=True) # Last update
timestamp

```

```

class BattleSession(models.Model):
    """
    Represents a real-time idiom battle session
    """
    player_1_id = models.ForeignKey(User, on_delete=models.PROTECT)
    player_2_id = models.ForeignKey(User, on_delete=models.PROTECT)
    start_time = models.DateTimeField() # When the game started
    winner_id = models.ForeignKey(User, on_delete=models.SET_NULL,
null=True)
    player_1_score = models.IntegerField() # Score of player 1
    player_2_score = models.IntegerField() # Score of player 2
    player_1_time = models.IntegerField() # Time to complete for player 1
(in seconds)
    player_2_time = models.IntegerField() # Time to complete for player 2
(in seconds)
    status = models.CharField(max_length=20) # Status of the session
(e.g., "completed", "cancelled")

```

```

class TestSession(models.Model):
    """
    Stores the details of a user's self-testing session
    """
    user_id = models.ForeignKey(User, on_delete=models.PROTECT)
    start_time = models.DateTimeField() # When the self-test started
    end_time = models.DateTimeField(null=True, blank=True) # When the
self-test ended
    score = models.IntegerField() # User's score in the session (Number of
correct answers)

```

```

class Question(models.Model):
    """
    Stores game questions and answers.
    """

```

```

question = models.CharField(max_length=100) # Question content
answer = models.CharField(max_length=100) # Correct answer
option_1 = models.CharField(max_length=100) # First answer option
option_2 = models.CharField(max_length=100) # Second answer option
option_3 = models.CharField(max_length=100) # Third answer option
option_4 = models.CharField(max_length=20, default="I don't know") #
Fourth answer option

```

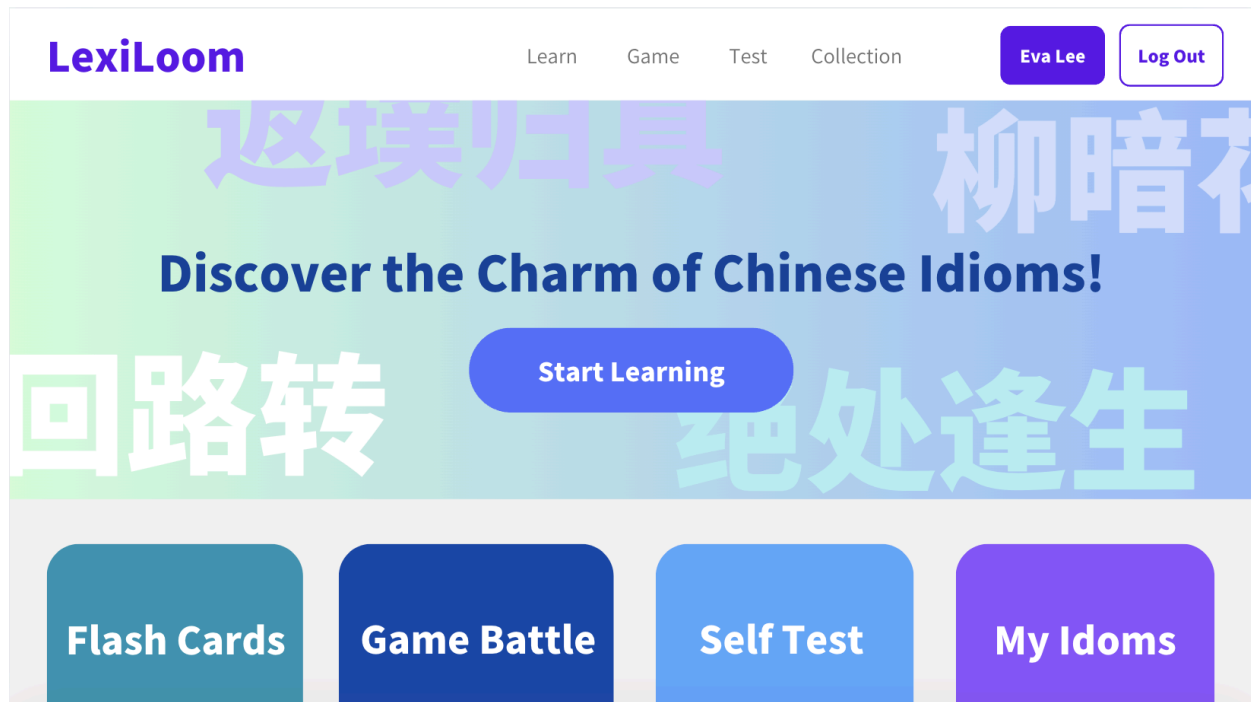
```

class Collection(models.Model):
    """
    Stores user-collected idioms.
    """
    user_id = models.ForeignKey(User, on_delete=models.PROTECT)
    idiom_id = models.ForeignKey(Idiom, on_delete=models.PROTECT)

```

5. Web MockUps

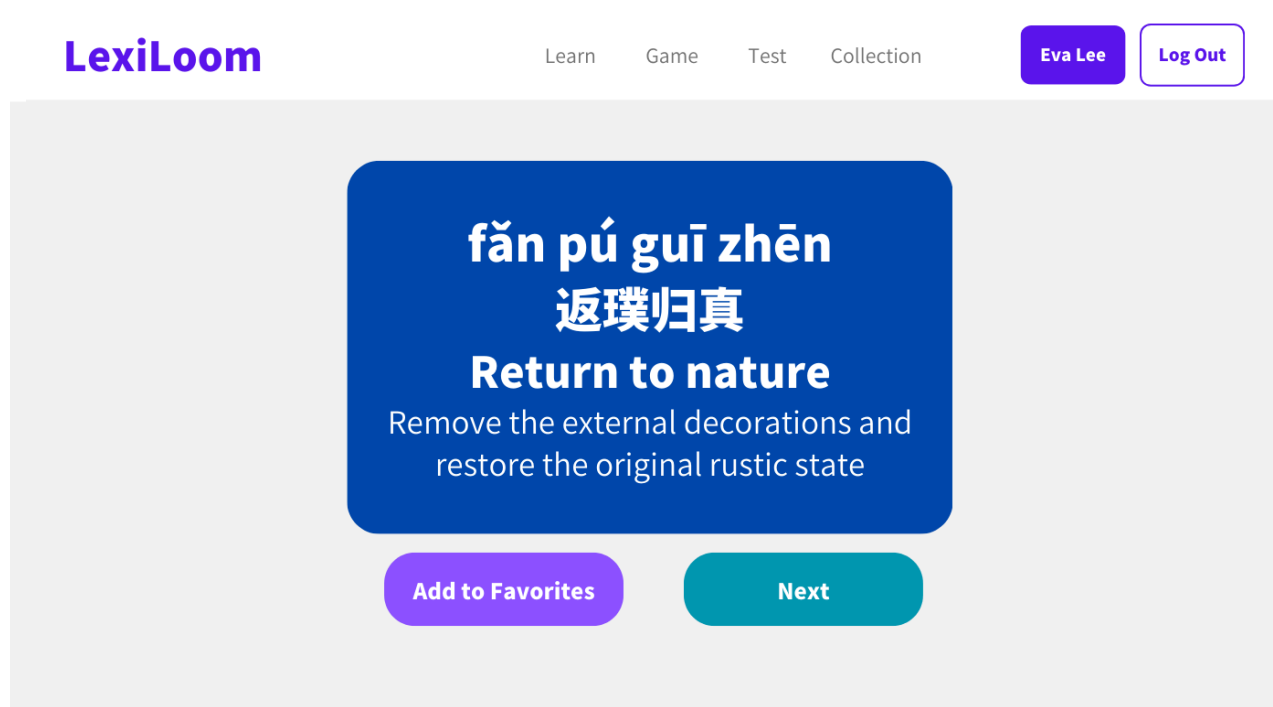
1. Home Page



Users can choose to start learning, self-testing, or start a game, etc.

2. Flash Cards

When the user clicks on the card, the card flips over, from only Chinese idioms on the front to English explanations on the back. Users can add idioms to favorites and proceed to the next card.



After a few studies, study cards will pop up. After the user selects an answer, the correct answer is displayed. Clicking on the card flips it over to show the explanation

返璞归真

Return to nature

Strive for success

Adapt and evolve

Pursue perfection

Add to Favorites

Next

3. Idiom Search

The user can input the English meaning of the desired idiom. The AI model called by the backend will generate some Chinese idioms and interpretations based on the user's input and display them to the user.

Get Your Idiom

Describe the idiom you are looking for!

Search

AI responses

4. My Idioms

Users can see the idioms they collect while learning, including their pinyin and English explanations.

My Idioms

fǎn pú guī zhēn
返璞归真

**Return to
nature**

Remove the external decorations and restore the original rustic state

fǎn pú guī zhēn
返璞归真

**Return to
nature**

Remove the external decorations and restore the original rustic state

fǎn pú guī zhēn
返璞归真

**Return to
nature**

Remove the external decorations and restore the original rustic state

5. Game Battle

A user can compete with a random player in a timed challenge. During the game, a player can see his own and the opponent's progress and scores, and the time left for answering the current question. All questions are single choice.


LexiLoom[Learn](#)[Game](#)[Test](#)[Collection](#)

Eva Lee

Log Out


Time Left: 15

Game Battle

510 (me)

50 : 30

Nick



A sudden turn of events where a desperate situation is unexpectedly resolved, often due to an unforeseen factor.

柳暗花明 (liǔ àn huā míng)

绝处逢生 (jué chù féng shēng)

峰回路转 (fēng huí lù zhuǎn)

起死回生 (qǐ sǐ huí shēng)

After the battle finishes, a user can see whether he wins or loses, as well as his own and the opponent's final scores. Then, the user can choose whether to return to the home page or begin a new battle.


LexiLoom[Learn](#)[Game](#)[Test](#)[Collection](#)


Eva Lee

Log Out

Game Battle

You win the game!

510's Score
100

Nick's Score
80

Return to Home

Next Battle

6. Self Test


This page is similar to the game battle, but it's designed for personal practice without an opponent.

LexiLoom

LearnGameTestCollection

Eva LeeLog Out

Self Test

510 (me)

50

A sudden turn of events where a desperate situation is unexpectedly resolved, often due to an unforeseen factor.

柳暗花明 (liǔ àn huā míng)

绝处逢生 (jué chù féng shēng)

峰回路转 (fēng huí lù zhuǎn)

起死回生 (qǐ sǐ huí shēng)

Organize files within your team repository however you want to organize the files for your overall project but place any specification—specific documents (such as your product backlog and first sprint backlog) in the root directory of your team's repository.