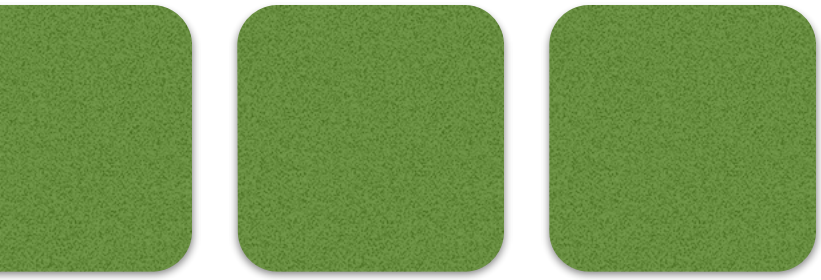


Pangaea

Jonathan Batscha
Eben Bitonte
Faruk Parhat
Guillermo Vargas

Uniting the world through language

Motivation

Pangaea aims to teach new languages to users through interaction with native speakers. In particular the application is designed to match people of different native languages so that they can exchange knowledge about their respective languages. Pangaea will connect matched users via a chat client, in which they will be able to engage in conversation in a language known by at least one user.

Individual Attention

This style of learning will provide one-on-one attention, and will expose users to the elements of language that are most relevant to contemporary communication.

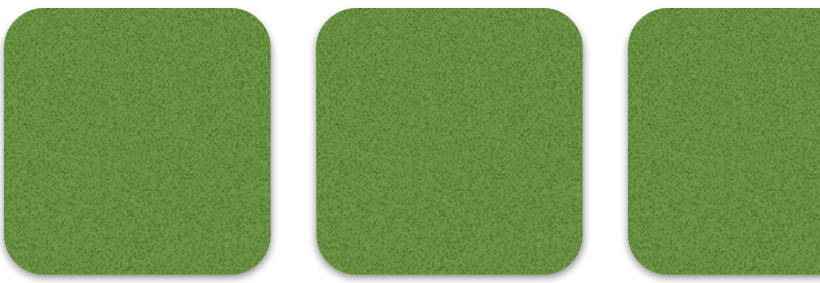
Casual Learning Environment

The casual nature of a chat client conversation should allow users to practice the conversational aspect of language, without focusing on the technical details associated with learning a new language.

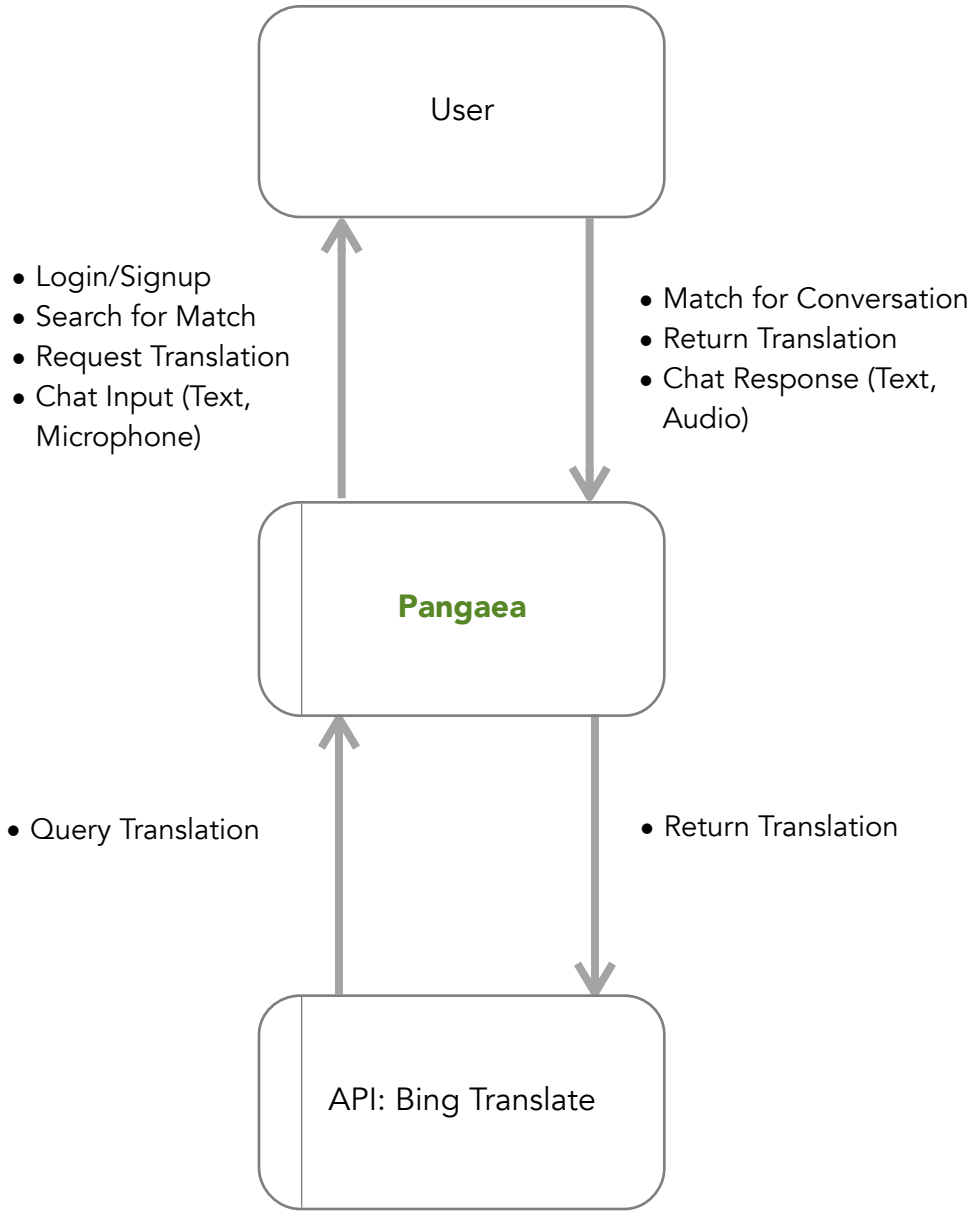
Contemporary Language

It is common for people to only want to learn the parts of a language necessary to communicate in everyday conversation. Pangaea aims to offer this by providing a medium for users to engage in a casual dialogue with users who understand the language as it is used today.

Languages today are mostly taught through grammar drills and vocabulary memorization. Such strategies entail tedious work, and seek to replace the human element of language with rote learning. This is effective if the learner's end goal is to read or write literature, or needs to learn all of the subtleties of the new language, however this is often not the case. Pangaea intends to develop the conversational skill necessary to comfortably visit a region that speaks a particular language. This will be accomplished through a common chat with another user, a medium that is more akin to real language usage than textbook study.



Context Diagram



Concepts

A Citizen

A **citizen** is a user of Pangaea. Each citizen has a list of **proficiencies**, the languages in which they are fluent. A citizen also has a username and a password. A citizen can request a chat to practice a given language. Two citizens will then be matched up according to their proficiencies and studies, thus initiating an **exchange**.

An Exchange

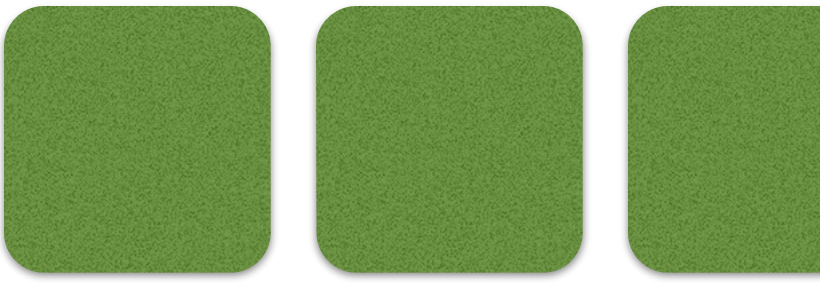
An **exchange** an ongoing conversation between 2 users. An exchange begins when 2 users first connect, and persists through multiple chat sesstion. A key purpose of Pangaea is to provide a platform for immersion into learning a language with another human being, and for this reason only supports one exchange at a time.

A Proficiency

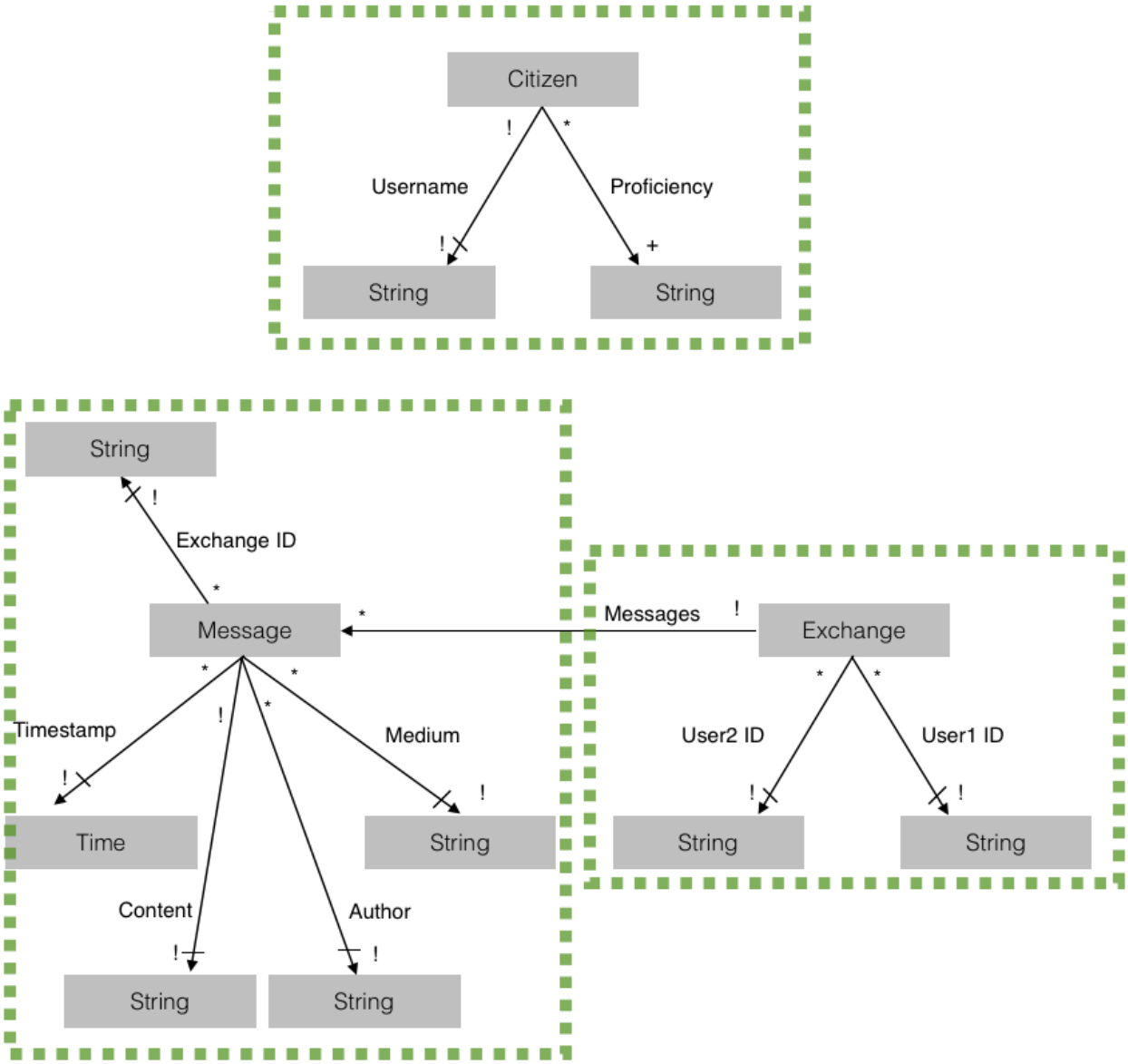
A **proficiency** is a language in which a **citizen** is fluent. The proficiency list can contain any number of languages, but we require users to pass a small quiz in order to add a language to their list.

A Study

A **study** is the language that a citizen is hoping to learn or use during the next exchange. A **citizen** chooses a language as his or her study before requesting a chat and therefore establishing an **exchange**, allowing for the ability to choose a different study before every exchange (though we anticipate most users will use Pangaea for a single language). This allows for the flexibility to change what language is currently being learned every time a user starts a new session, while making it easier for students of a single language to type a the name of a given language repeatedly.



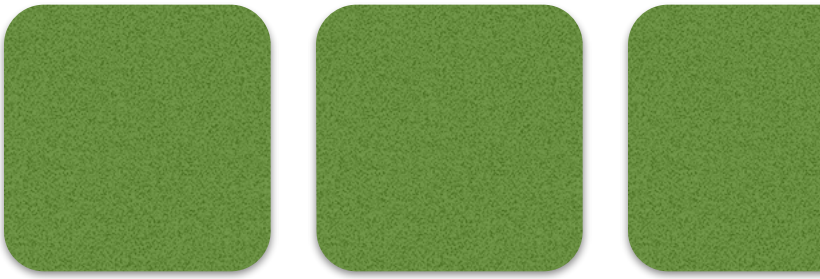
Data Model Diagram (based on lecture 9)



Data Design Example (based on lecture 9)

```
Citizen
{
Username: “Joe”,
 Proficiency: [“Spanish”,”English”]
}

Exchange
{
User1ID: “123”,
User2ID: “456”
Messages:[
     {
     Medium: “Text”,
     Author: “Bob”,
     Content: “Hola!”,
     Timestamp: “1416369281003”,
     ExchangeID: “507f1f77bcf86cd799439011”
     },
     {
     Medium: “Text”,
     Author: “Joe”,
     Content: “Muy bien”,
     Timestamp: “1416369281004”,
     ExchangeID: “507f1f77bcf86cd799439011”
     }
}
```

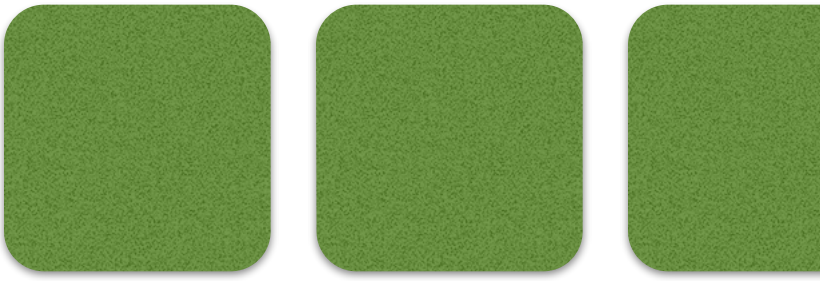


Data Model Description

Our data model consists of 3 main objects: Citizens, Exchanges, and Messages. Each user is referred to as a 'Citizen' and has 2 main attributes: a username (String), and a list of proficiencies (languages that they are proficient in). A Citizen communicates with another Citizen in an Exchange. An exchange consists of the two Citizens in the exchange (immutable), along with a list of 'Messages' (mutable) posted in the exchange by those 2 users. A Message consists of 5 attributes: an immutable indicator on whether message is audio or video; an author (immutable); content (immutable); a timestamp indicating when message was sent (immutable); and the ID of the Exchange in which message appears (immutable).

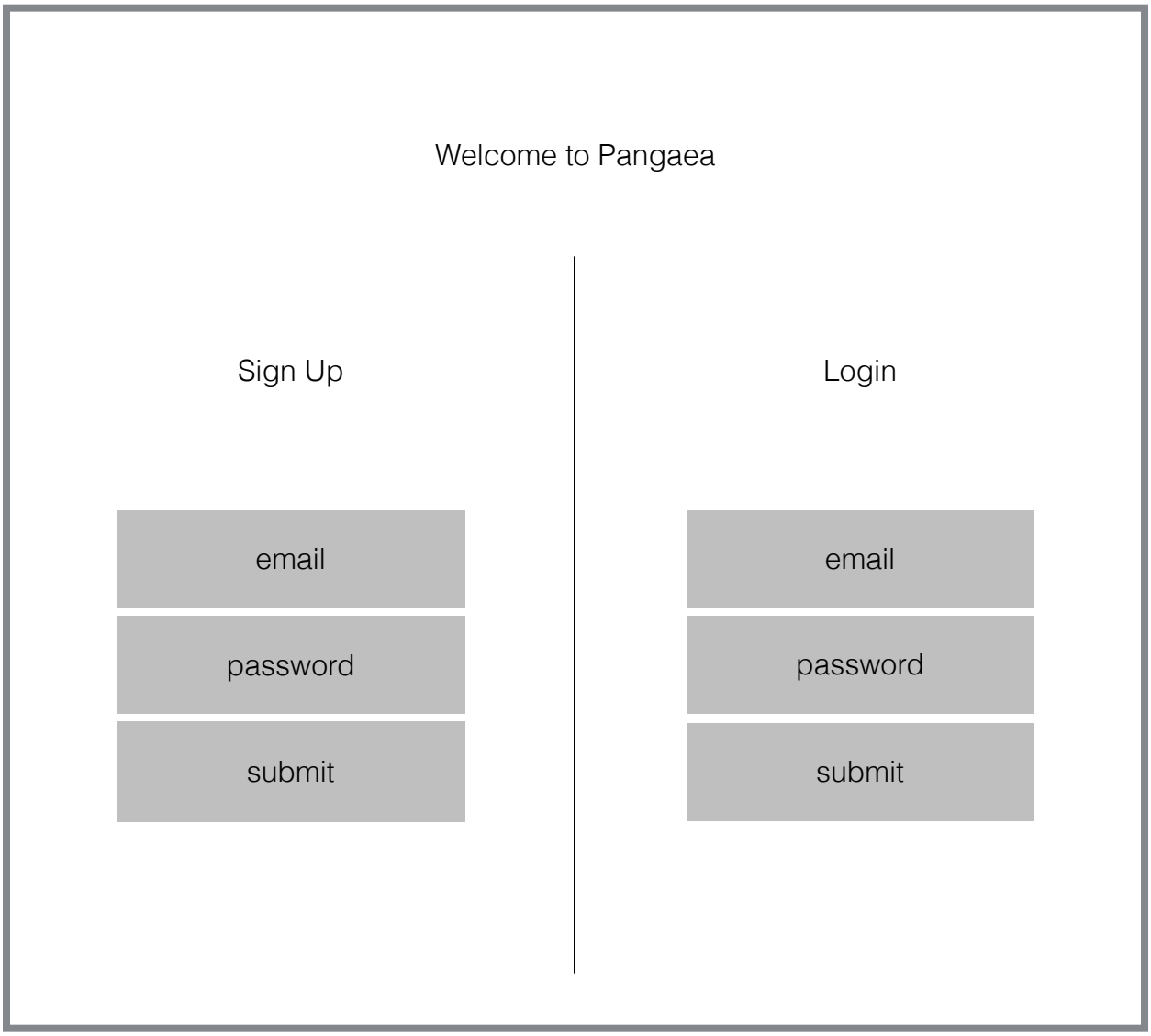
Security

Our main security concerns are in safeguarding the privacy of our users' information, along with preventing malicious users from attacking our site and otherwise diminishing the language-learning experience of our valued users. We plan to safeguard our users' information by requiring secure passwords, as well as bolstering this security by making our code immune to common attacks such as those that fall under "injection hacks" and "cross-site scripting". A more complicated decision is in how we deal with spammers and other malicious users who negatively impact the experience of other users. We have thought of a few approaches to mitigate this concern. One method to prevent repeat-offenders from negatively affecting others is by allowing users to report a malicious user. Perhaps a more involved way to handle this is by including a rating system, whereby users get rated on their quality of conversation; we can use this information to give users higher ratings greater preference in getting matched to chat with another user, whether or not they are the ones requesting a chat. Another idea we had in preventing users from claiming that they know a language that they don't is via a "test-captcha", in which we present the user with many similar sentences, and the user must select the one that is grammatically correct, though if we can successfully screen for malicious users in other ways, such a test may be an unnecessary burden on our intended users. The thinking behind this is if we do go ahead with a rating system, that rating system will already account for whether or not a given user knew a language that they listed as a proficiency.



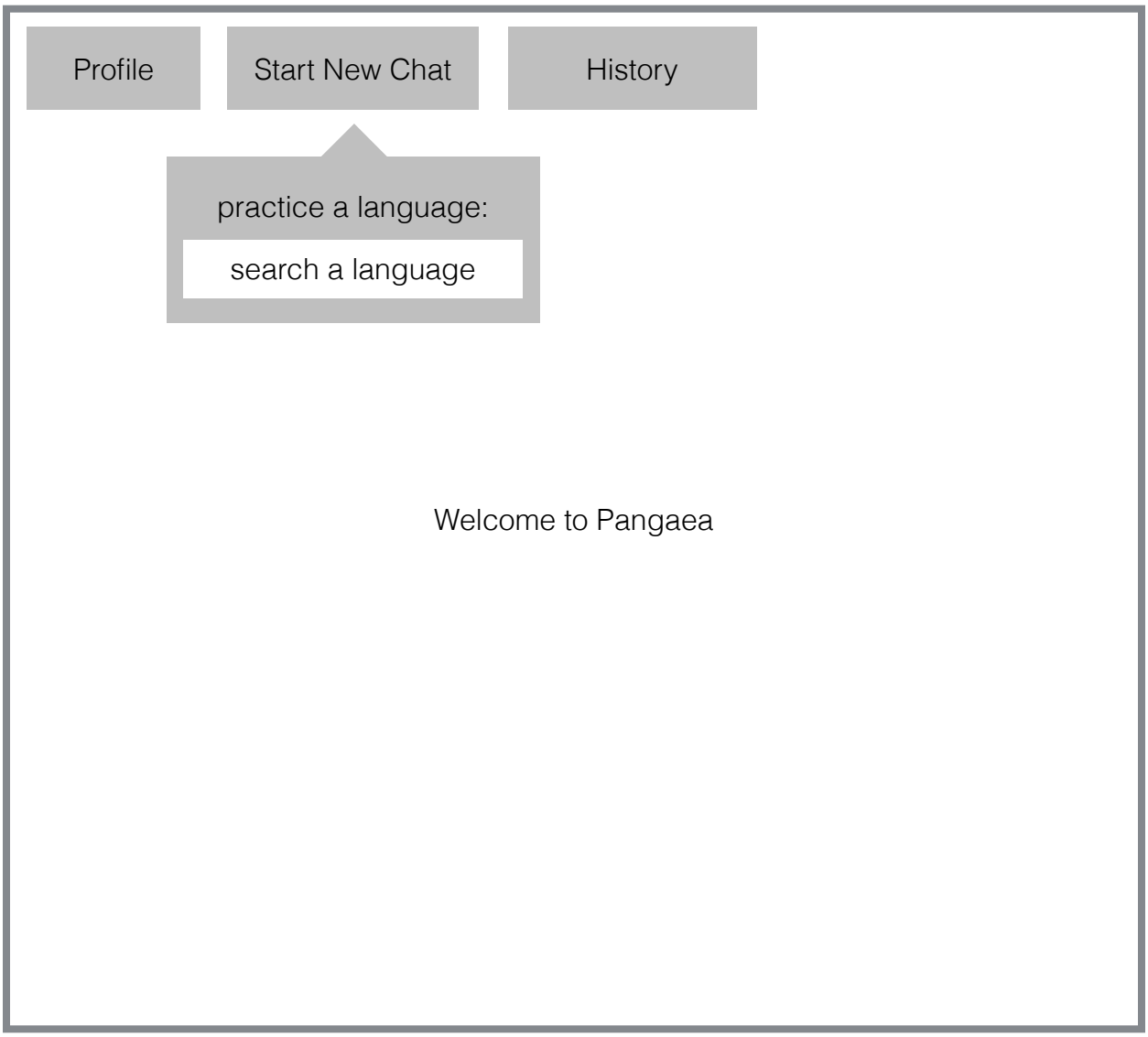
User Interface Wireframes

Welcome Screen (not logged in)



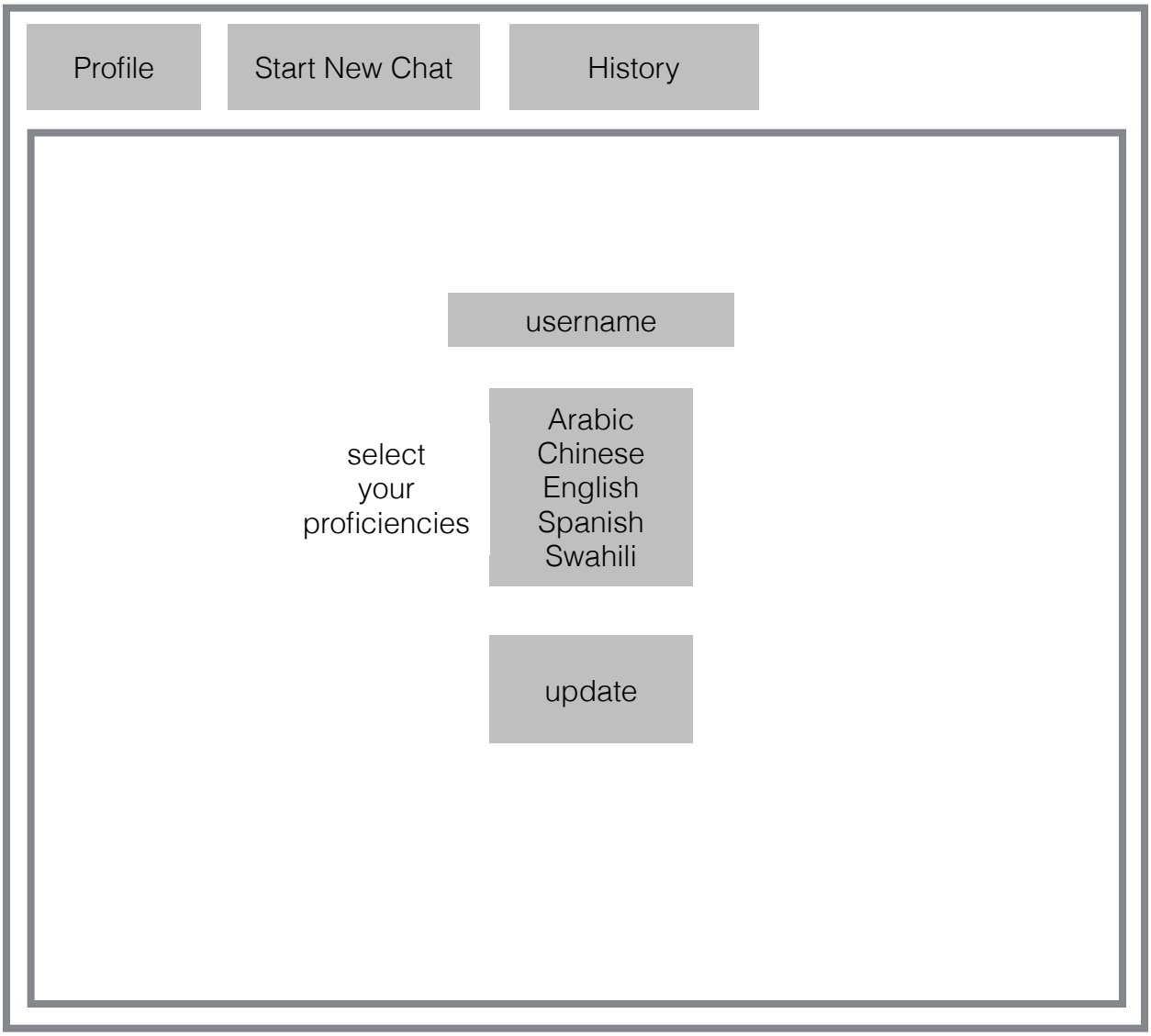
User Interface Wireframes (cont'd)

Welcome Screen (logged in)



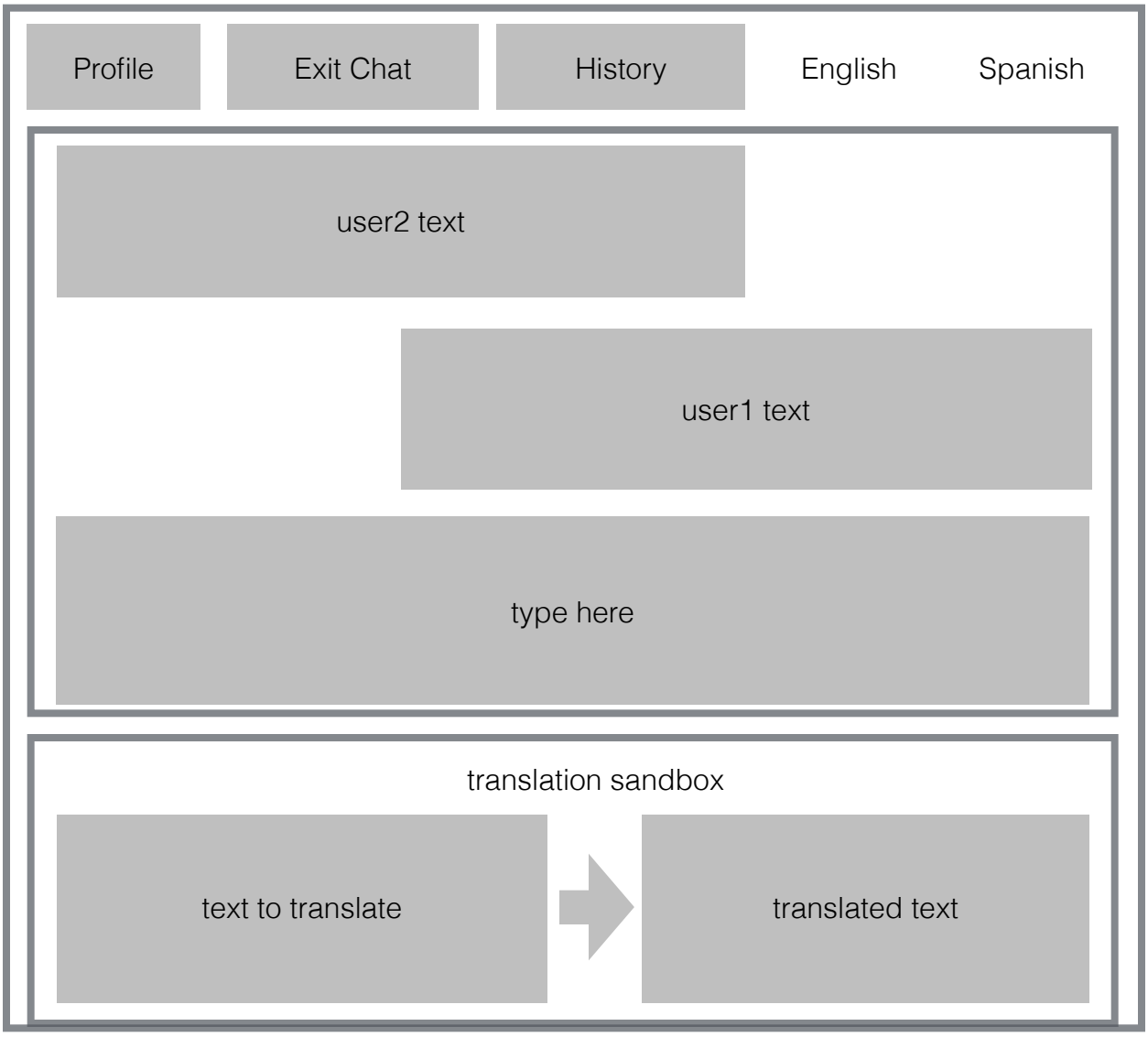
User Interface Wireframes (cont'd)

Profile Screen



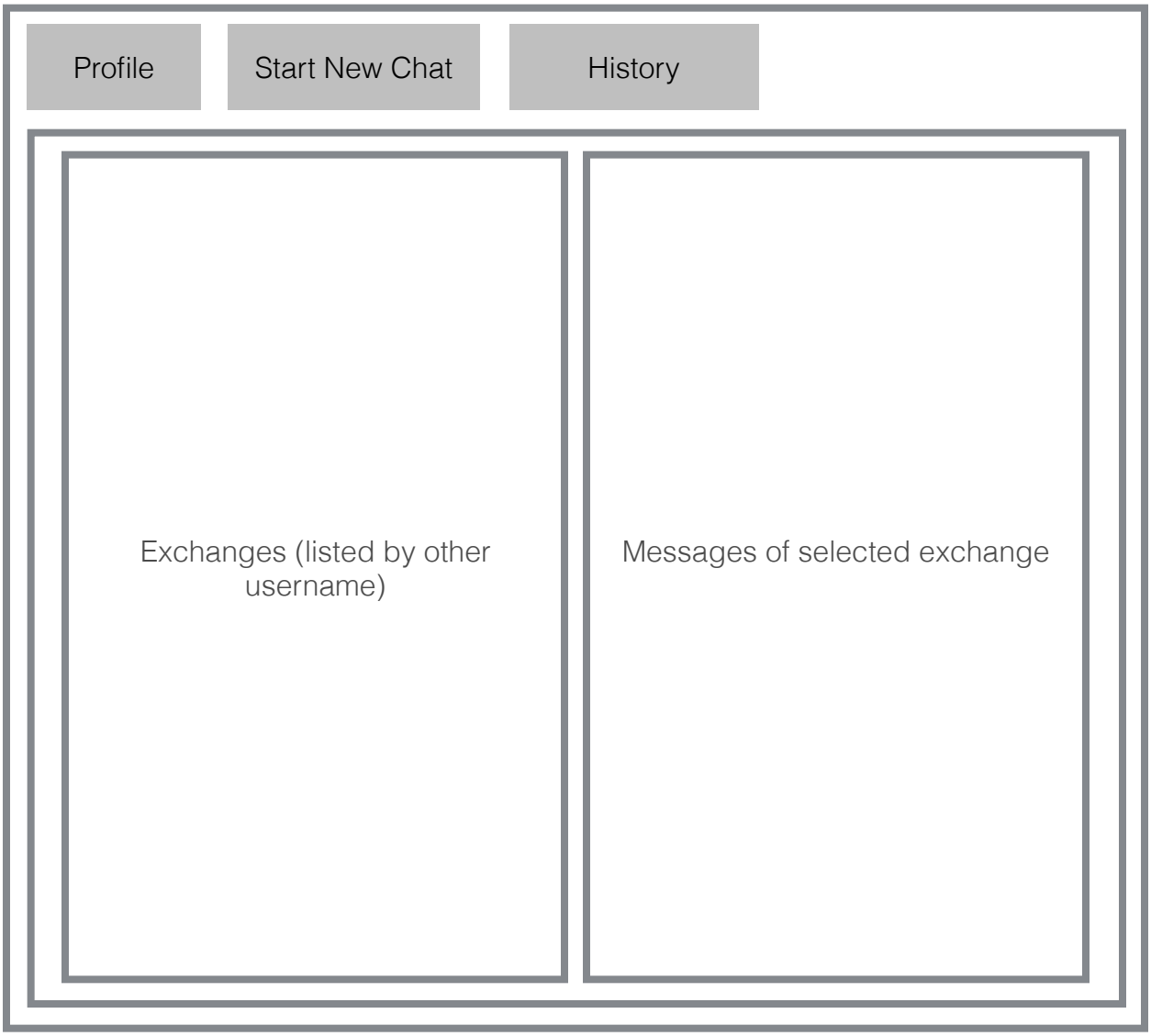
User Interface Wireframes (cont'd)

Chat Screen



User Interface Wireframes (cont'd)

Activity Screen



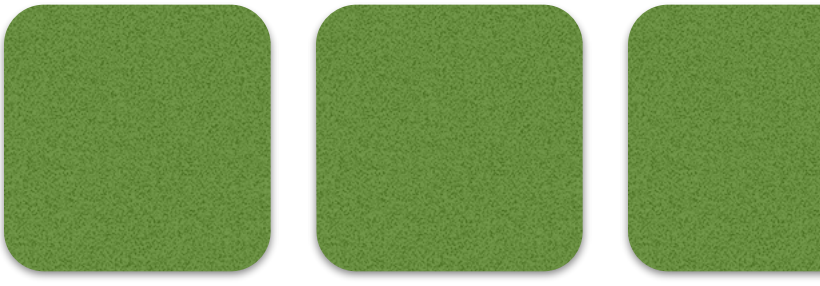
Design Challenges

We faced a quite a few challenges in designing our MVP, both in revisiting our design and improving it, as well as in implementing it in the MVP. We will begin be discussing the challenges we faced in revisiting our original design, and then go on to discussing the challenges we faced in implementing our design in the MVP (many).

From our initial pitch, team discussion, team plan, and other work prior to our first meeting with out mentors, it was clear we were all very passionate about our Pangaea. However, as we began to delve into the details, it was clear there were quite a few small yet crucial design decisions that we needed to make, about which we all contributed our own opinions.

One such decision was how users should initiate a chat: should they have a list of languages they are studying, one of which they can choose, or should they have no such list, and rather have to search for a language in order to start a chat in that language. We weighed the pros and cons of both. Saving languages a citizen is studying helps us create a "perfect" match (i.e Spanish speaker learning english with english speaker learning language); these are considered optimal, as both users have an incentive to continually seek each other out and chat over multiple sessions; further, for users only studying one language (say, Spanish), they can start a chat in a single click, without having to type in Spanish (though this downside will be mitigated by using autofill). On the other hand, omitting that attribute from citizen profiles results in a quicker setup and simpler design for new users. Weighing these two options, we opted to go with the second option: omitting the languages studied from citizen profiles, and instead initiating chat in a given language via a search bar.

Another such design challenge was whether chats should be logged, and if so, should each chat session be logged separately, or should all chats between a pair of users be stored in a single continuous log. To answer the first question, we thought forward to how we would like to see users learn on our site, and we decided that users would get the most out of our site if they can review a cool conversation that they had, and decided that we should definitely log chats. The next question was a little less clear, at least at first. Some of us were of the opinion that storing separate chat sessions in separate logs is preferable, as that way, users could go back to a specific chat where they learned something cool; some of us on the other hand thought that this added little functionality for the non-trivial amount of complexity we would be adding, both coding-wise and in terms of aesthetics. Thus, we decided that for our MVP, we would go with the simpler of the two options (saving all chats between two users in a single, continuous log), and that we would revisit the other option after the MVP.



Design Challenges (cont'd)

Now on the topic of design challenges that we encountered while implementing the MVP….
(to be continued later)