

Elsbeth Turcan and Katie Stasaski
CAPITAL Passages and reMind

Project Goals

There are two components to this project: an educational system called reMind and an automatic question-generation tool called CAPITAL Passages. The primary goal of reMind is to provide instructors and students with a platform for instructors to create learning materials and students to complete them. Because instructors will be able to share questions they have created, this system will be collaborative and increase the quality of questions as well as the speed with which instructors can assemble course materials. Students will be able to complete the exercises with an HTML5 application, meaning that they will be able to practice their materials conveniently anywhere their mobile device or browser can access the Internet. Meanwhile, CAPITAL Passages, an add-on for reMind, aims to generate reading comprehension questions from an input passage of English text in order to provide faster and easier ways for instructors to create materials for students learning to read at the semantic level.

Users and User Interactions

There are four types of users in the reMind and CAPITAL Passages system.

1) SUPER ADMINISTRATOR

A super administrator is a sort of omnipotent manager for the reMind system, with the ability to edit almost any data in the system. There will be only a few super administrators. They will do things like register institutions and their administrators.

2) ADMINISTRATOR

An administrator is a manager for a single reMind-using institution (an institution being an organization or collection of instructors and students—for example, one school might be one institution). They will manage the instructors and students at their particular institution. Each institution can have one or more administrators.

3) INSTRUCTOR

An instructor is an individual teaching a course and providing or putting together learning and testing materials for students. Instructors will belong to one institution and manage their courses at that institution. An instructor is also the only user who will have access to CAPITAL Passages and be able to generate comprehension questions from passages.

4) STUDENT

A student is a user who is using the reMind system to learn. Students will belong to a certain institution, enroll in courses, and most importantly complete material prepared by instructors in their courses. Students will answer the questions generated by CAPITAL Passages; this will be the extent of their interaction with Passages, but their correctness statistics and responses will help improve the questions Passages generates.

Each user can perform a variety of actions described in detail below. A super administrator has all the privileges of an administrator, who in turn has all the privileges of an instructor. Any user must be logged into the system to perform any action.

Model Descriptions

This project uses a model-view-controller framework. To describe the use cases, we need to define a few models in the system. This description is somewhat below the component level detailed later, but is important to understanding the users' actions.

Institution – A single organization using the reMind system, such as one single school. An institution has a set of users associated with it, including administrators, instructors, and students. Courses are offered by a single institution.

Course – A collection of modules. Each course has a start and end date during which students enrolled in that course can access it. New modules can be added at any time. A single instructor or administrator teaches a course, meaning that they manage the material students learn therein.

Module – A collection of questions. Each module can be activated, which means that it is made available to students and it can no longer be edited. A module is analogous to the idea of a lesson or chapter of material. A module can be of three types: practice, survey, or evaluation. A practice module can be repeated any number of times; a survey module can be taken only once and is intended for data collection purposes; an evaluation module can be taken only once and is intended for instructors to assess students' progress.

Question – The smallest unit with which students can interact. A question has a prompt and requires a response from a student user.

Workflow Types

Most of the actions taken by users in this system follow one of several workflow patterns:

Database query (website) – The user, on the reMind website, clicks some button requesting to view some data—such as all courses they are teaching or all questions in the public question pool. A query is sent by Play Framework to the MySQL database, and the results of the query are transmitted back and presented to the user on a reMind webpage.

Database query (mobile application) – The user, on the reMind mobile application, presses some button requesting to view some data—such as all courses they are enrolled in or the next question in their current practice module. A query is sent through Play Framework to the MySQL database, and the results of the query are transmitted back and presented to the user on an HTML5 reMind page.

Database submission (website) – The user, on the reMind website, clicks some button indicating that they wish to make some simple change to their data—such as to activate a module. This action may submit new records into the database or overwrite existing ones. The query is sent by Play Framework to the MySQL database and it adds or overwrites the proper entries in the database; the reMind website shows the user some indication as to whether their actions were successful.

Database submission (website, using form) – The user, on the reMind website, fills out some form with some data they wish to input—such as the details of a new course or question—and clicks a button to submit it. This action may submit new records into the database or overwrite existing ones (such as to edit an existing course). The query is sent by Play Framework to the MySQL database and it adds or overwrites the proper entries in the database; the reMind website shows the user some indication as to whether their actions were successful.

Database submission (mobile application) – The user, on the reMind mobile application, presses some button indicating that they wish to make some simple change to their data—such as to enroll in a course. This action may submit new records into the database or overwrite existing ones. The query is sent through Play Framework to the MySQL database and it adds or overwrites the proper entries in the database; the reMind mobile application shows the user some indication as to whether their actions were successful.

Database submission (mobile application, using form) – The user, on the reMind mobile application, fills out some form with some data they wish to input—specifically, this use case is for answering questions, which includes pressing buttons for correct answers or filling in text—and presses a button to submit it. This action submits new records into the database. The query is sent by Play Framework to the MySQL database and it adds the proper entries in the database.

Database deletion (website) – The user, on the reMind website, clicks some button indicating that they wish to remove some information—such as to delete an announcement or reject a student’s submitted question. The query is sent by Play Framework to the MySQL database, where it does not actually remove any records, but sets a “disavowed” flag in each record to true, meaning that it will no longer be displayed. This is for ease of retrieval for accidentally-deleted data. The reMind website shows the user some indication as to whether their actions were successful.

Database deletion (mobile application) – The user, on the reMind mobile application, clicks some button indicating that they wish to remove some information—such as to reject another student’s rivalry request. The query is sent through Play Framework to the MySQL database, where it does not actually remove any records, but sets a “disavowed” flag in each record to true, meaning that it will no longer be displayed. This is for ease of retrieval for accidentally-deleted data. The reMind website shows the user some indication as to whether their actions were successful.

Use Cases/User Interactions

1) SUPER ADMINISTRATOR

1. **Direct database manipulation** – A super administrator's role is highly managerial, and they are afforded a lot of power over the system. Therefore, a super administrator's unique use cases all involve and boil down to directly manipulating MySQL databases: they can add multiple students to the system, add administrators to the system, restore accidentally deleted data, and so on.

1. *Workflow:* Direct database manipulation. The super administrator will access the MySQL database directly and make queries, submissions, and deletions using MySQL.

2. *Functional components*: MySQL database

2) SUPER ADMINISTRATOR AND ADMINISTRATOR

1. **View instructors** – An administrator may choose to view all instructors registered under their institution.
 1. *Workflow*: Database query (website)
 2. *Functional components*: reMind website, MySQL database
2. **Add an instructor** – An administrator may create a new instructor to be registered under their institution. The new instructor's username must not already be in use.
 1. *Workflow*: Database submission (website, using form)
 2. *Functional components*: reMind website, MySQL database
3. **Edit an instructor** – An administrator may edit the information of an instructor registered under their institution, including their name, username, and password. If the administrator chooses a new username, the new username must not already be in use.
 1. *Workflow*: Database submission (website, using form)
 2. *Functional components*: reMind website, MySQL database
4. **Delete an instructor** – An administrator may delete an instructor registered under their institution.
 1. *Workflow*: Database deletion (website)
 2. *Functional components*: reMind website, MySQL database
5. **Add a student** – An administrator may create a new student to be registered under their institution. The new student's username must not already be in use.
 1. *Workflow*: Database submission (website, using form)
 2. *Functional components*: reMind website, MySQL database
6. **Edit a student** – An administrator may edit the information of a student registered under their institution, including their name, username, and password. If the administrator chooses a new username, the new username must not already be in use.
 1. *Workflow*: Database submission (website, using form)
 2. *Functional components*: reMind website, MySQL database
7. **Delete a student** – An administrator may delete a student registered under their institution.
 1. *Workflow*: Database deletion (website)
 2. *Functional components*: reMind website, MySQL database
8. **Edit a course's instructor** – An administrator may change which instructor is assigned to a course currently being taught at their institution. They will only be able to select from instructors who are registered under their institution.
 1. *Workflow*: Database submission (website, using form)
 2. *Functional components*: reMind website, MySQL database

3) SUPER ADMINISTRATOR, ADMINISTRATOR, AND INSTRUCTOR

1. **Edit profile** – An administrator or instructor may edit their own profile information, including a profile photo, name, username, and password. If they edit their own username, the new username must not already be in use.

1. *Workflow*: Database submission (website, using form)
 2. *Functional components*: reMind website, MySQL database
2. **View courses** – An administrator or instructor may view courses. Administrators may view all courses created at their institution, while instructors may only view the courses they are teaching.
 1. *Workflow*: Database query (website)
 2. *Functional components*: reMind website, MySQL database
3. **Add a course** – An administrator or instructor may create a new course they intend to teach at their institution
 1. *Workflow*: Database submission (website, using form)
 2. *Functional components*: reMind website, MySQL database
4. **Edit a course** – An administrator or instructor may edit the details of a course, including name and start and end dates. Administrators may edit any course in their institution, while instructors may only edit a course they are teaching.
 1. *Workflow*: Database submission (website, using form)
 2. *Functional components*: reMind website, MySQL database
5. **Delete a course** – An administrator or instructor may delete a course. Administrators may delete any course in their institution, while instructors may only delete a course they are teaching. A course cannot be deleted by an administrator or instructor between its start and end date; only a super administrator can delete a course then through database manipulation.
 1. *Workflow*: Database deletion (website)
 2. *Functional components*: reMind website, MySQL database
6. **View modules** – An administrator or instructor may view the modules in any course. Administrators may view modules for any course in their institution, while instructors may only view the modules of a course they are teaching.
 1. *Workflow*: Database query (website)
 2. *Functional components*: reMind website, MySQL database
7. **Create a module** – An administrator or instructor may create a new module in any course. Administrators may add a new module to any course in their institution, while instructors may only add a new module to a course they are teaching.
 1. *Workflow*: Database submission (website, using form)
 2. *Functional components*: reMind website, MySQL database
8. **Delete a module** – An administrator or instructor may delete a module from any course. Administrators may delete a module from any course in their institution, while instructors may only delete a module from a course they are teaching. A module cannot be deleted by an administrator or instructor after it has been activated but before its end date; only a super administrator can delete a module then through database manipulation.
 1. *Workflow*: Database deletion (website)
 2. *Functional components*: reMind website, MySQL database
9. **Edit a module** – An administrator or instructor may edit a module in any course. Administrators may edit a module in any course in their institution, while instructors may only edit a module in a course they are teaching.

1. *Workflow*: Database submission (website, using form)
 2. *Functional components*: reMind website, MySQL database
10. **Activate a module** – An administrator or instructor may activate a module in any course; after a module is activated, it may no longer be edited, and students may view it. Administrators may activate a module in any course in their institution, while instructors may only activate a module in a course they are teaching.
 1. *Workflow*: Database submission (website)
 2. *Functional components*: reMind website, MySQL database
11. **View students** – An administrator or instructor may choose to view all students registered under their institution.
 1. *Workflow*: Database query (website)
 2. *Functional components*: reMind website, MySQL database
12. **Accept or reject student enrollment requests** – For a course they teach, an administrator or instructor may view all students who have requested enrollment in that course, and accept or reject each request.
 1. *Workflow*: Database submission (website), Database deletion (website)
 2. *Functional components*: reMind website, MySQL database
13. **Enroll a student in a course** – An administrator or instructor may manually enroll single students in any course. Administrators can enroll students in any course in their institution, while instructors may only enroll students in a course they are teaching.
 1. *Workflow*: Database submission (website)
 2. *Functional components*: reMind website, MySQL database
14. **Unenroll a student from a course** – An administrator or instructor may manually unenroll single students from any course. Administrators can unenroll students from any course in their institution, while instructors may only unenroll students from a course they are teaching.
 1. *Workflow*: Database deletion (website)
 2. *Functional components*: reMind website, MySQL database
15. **Make an announcement** – An administrator or instructor may issue an announcement to any course. Administrators can issue announcements to any course in their institution, while instructors may only issue announcements to a course they are teaching.
 1. *Workflow*: Database submission (website, using form)
 2. *Functional components*: reMind website, MySQL database
16. **Accept or reject a question made by a student** – An administrator or instructor may view all questions that students have created and submitted for any course, and accept or reject them. Administrators can view submitted questions for any course in their institution, while instructors may only view submitted questions for a course they are teaching.
 1. *Workflow*: Database submission (website, using form), Database deletion (website)
 2. *Functional components*: reMind website, MySQL database
17. **Create a survey** – An administrator or instructor may create a survey module, which contains questions intended to gather information from students, for any course.

Administrators can create survey modules in any course in their institution, while instructors may only create survey modules in a course they are teaching.

1. *Workflow*: Database submission (website, using form)
 2. *Functional components*: reMind website, MySQL database
18. **View student progress** – An administrator or instructor may view the progress of all students in any course. Administrators can view student progress in any course in their institution, while instructors may only view student progress in a course they are teaching.
1. *Workflow*: Database query (website)
 2. *Functional components*: reMind website, MySQL database
19. **Create a question** – An administrator or instructor may create a new question for a module in any course. Administrators can create questions in any course in their institution, while instructors may only create questions in a course they are teaching.
1. *Workflow*: Database submission (website, using form)
 2. *Functional components*: reMind website, MySQL database
20. **Browse public questions** – An administrator or instructor may view the pool of questions marked by other administrators and instructors as public when adding questions to a module. They may select any question and add it to the module they are currently creating.
1. *Workflow*: Database query (website)
 2. *Functional components*: reMind website, MySQL database
21. **Generate questions from a passage** – An administrator or instructor may choose to use Passages to generate reading comprehension questions. They may input their own passage or choose from passages already stored in the database. These questions will act exactly the same as normal questions described above once generated.
1. *Workflow*: This workflow runs through the Passages system. On the reMind website, the user either pastes a passage into a text box or clicks a button to browse passages in the database (a website database query). The user selects a passage from either source and confirms that they want to generate questions by clicking a button. The passage is sent to the CAPITAL Passages generator, which generates questions according to its algorithms and sends them back to the website as Question objects for the user to view them.
 2. *Functional components*: reMind website, MySQL database, Passages generator
22. **Accept or reject Passages-generated questions** – When creating Passages questions, an administrator or instructor may view the questions generated by Passages and choose to accept or reject individual questions. They may also re-generate questions to show new questions as many times as they desire.
1. *Workflow*: This workflow runs through the Passages system. The user views all questions generated by Passages and clicks a button to remove each question they do not like. If they desire, the user may click a button to generate new questions from the same passage, which loops back into the previous use case, using the current passage as input.
 2. *Functional components*: reMind website, Passages generator

4) STUDENT

1. **Create a question** – A student may create their own question for any course in which they are enrolled and submit it to their instructor for approval.
 1. *Workflow*: Database submission (website, using form)
 2. *Functional components*: reMind mobile application, MySQL database
2. **Register at an institution** – A student may search for an institution and register at it. Their registration request will need to be reviewed by an administrator.
 1. *Workflow*: Database query (mobile application), Database submission (mobile application)
 2. *Functional components*: reMind mobile application, MySQL database
3. **Enroll in a course** – A student may browse courses at an institution under which they are registered and choose to enroll in any course. Their enrollment request will need to be reviewed by the administrator of their institution or the instructor of that course.
 1. *Workflow*: Database submission (mobile application)
 2. *Functional components*: reMind mobile application, MySQL database
4. **Unenroll from a course** - A student may choose to unenroll from any course in which they are currently enrolled.
 1. *Workflow*: Database deletion (mobile application)
 2. *Functional components*: reMind mobile application, MySQL database
5. **View courses** – A student may view all courses in which they are currently enrolled.
 1. *Workflow*: Database query (mobile application)
 2. *Functional components*: reMind mobile application, MySQL database
6. **Answer a question in the daily task** – For any course in which they are currently enrolled, a student may view a daily task each day, which consists of practice questions picked by a spaced repetition algorithm. The student sees this task when viewing a specific course for the first time each day, and they must complete it in order to practice other questions in that course. Upon completing a question in the daily task, a student is given the correct answer immediately.
 1. *Workflow*: Database query (mobile application), Database submission (mobile application, using form).
 2. *Functional components*: reMind mobile application, MySQL database
7. **Answer a question in a practice module** – For any course in which they are currently enrolled, a student may practice materials at will. The student must have completed their daily task. The student may only practice questions from a module that has been activated and has not yet reached its end date. Upon completing a question for practice, a student is given the correct answer immediately.
 1. *Workflow*: Database query (mobile application), Database submission (mobile application, using form). Once the user submits their answers the app will also highlight which of the answers was the correct answer.
 2. *Functional components*: reMind mobile application, MySQL database
8. **Answer a question in a survey or evaluation** – For any course in which they are currently enrolled, a student may answer survey or evaluation questions for any currently

active survey or evaluation module. Upon completing a question in one of these types of modules, the student is not shown the correct answer (for a survey module, correct answers do not even exist).

1. *Workflow*: Database query (mobile application), Database submission (mobile application, using form).
 2. *Functional components*: reMind mobile application, MySQL database
9. **View leaderboard** – For any course in which they are currently enrolled, a student may view other students' completion progress as compared to their own. A student can only view other students by name if they are their “rivals”; otherwise students will see anonymous statistics such as their percentile within the course.
1. *Workflow*: Database query (mobile application)
 2. *Functional components*: reMind mobile application, MySQL database
10. **Request a rivalry** – A student may choose to send a rivalry request to any other student in their institution, meaning that the students would be able to see each other's progress by name on the leaderboards. This request must be reviewed by the other student.
1. *Workflow*: Database submission (mobile application)
 2. *Functional components*: reMind mobile application, MySQL database
11. **Accept or reject a rivalry** – A student may receive a notification that another student has requested a rivalry and accept or reject a rivalry with that student.
1. *Workflow*: Database submission (mobile application), Database deletion (mobile application)
 2. *Functional components*: reMind mobile application, MySQL database

Functional Components

There are four main functional components in this system.

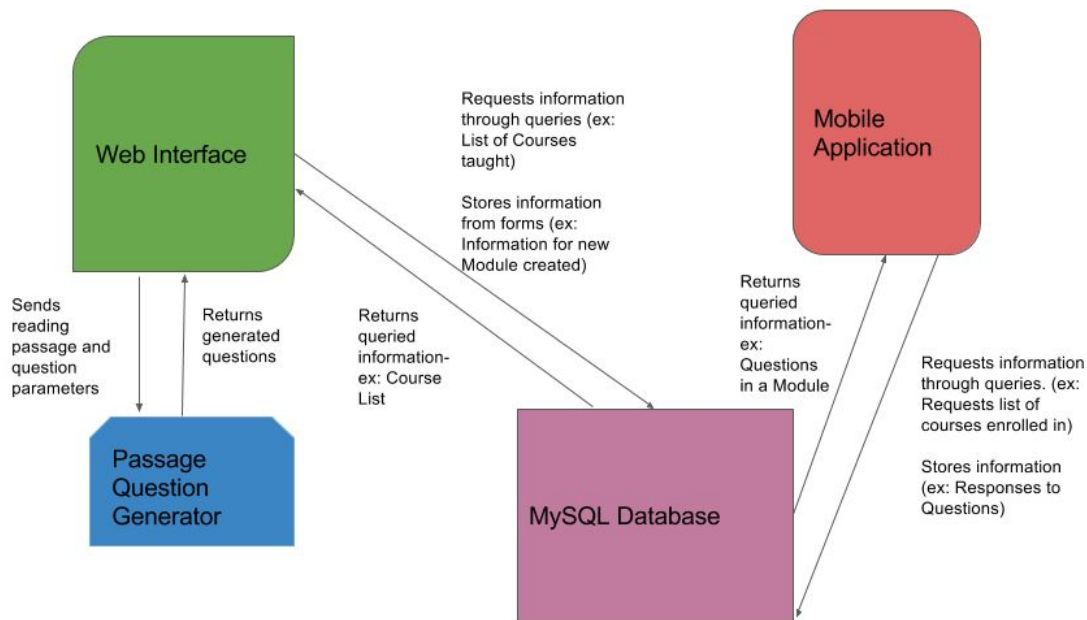
ReMind website – A website through which instructors can create course materials for students to practice and study. Most of the user-visible functionality will be contained in the website, which runs on Play Framework. For example, students can log onto the website and create questions, and instructors have access only to the website.

ReMind mobile application – An HTML5 mobile application through which students perform the majority of their tasks. Students can access this application using a mobile device or any browser.

Passages question generator – The backing algorithm for the automatic reading comprehension question generator. It will accept English-text passages and automatically generate several types of reading comprehension questions from them; these questions will be sent to the reMind system and treated exactly the same as any user-generated question.

MySQL database – The backing storage utility for the entire system. It will run on a server and store any data that the system requires, such as lists of users, lists of courses, student responses to questions, and so on. The typical user will never interact with the MySQL database.

Component Interaction



Functional and Non-Functional Requirements

	Functional	Non-Functional
ReMind Website	<ul style="list-style-type: none"> Allows user to log in Allows user to manage profile Allows user to manage courses Allows user to contribute to and view public questions Allows user to manage students Allows student user to submit questions for approval Allows access to CAPITAL Passages 	<ul style="list-style-type: none"> User-friendly UI Secure system (secure password storage) Allows recovery of deleted data Easily expanded functionality
ReMind Mobile Application	<ul style="list-style-type: none"> Allows user to log in Allows user to complete daily task 	<ul style="list-style-type: none"> User-friendly UI Secure system (secure password)

ReMind Mobile Application (cont'd)	<ul style="list-style-type: none"> • Allows user to practice questions • Allows user to answer survey questions • Allows user to answer evaluation questions • Allows user to view own and rivals' progress • Allows user to create a rivalry 	<ul style="list-style-type: none"> • storage) • Local storage of progress without Internet access • Multiple platform support (HTML5)
Passages Generator	<ul style="list-style-type: none"> • Generates reading comprehension questions from an English passage • Generates multiple types of questions based on input parameters 	<ul style="list-style-type: none"> • Reasonable speed
MySQL Database	<ul style="list-style-type: none"> • Stores all data in the system 	<ul style="list-style-type: none"> • N/A

***Note:** We are not building the MySQL database; it is simply a large tool we are using, and we are building our system on top of it.