**Bindr: Tinder for Studying**

*Kevin Cadavillo, Jack Wu, Feiying Zheng, Jake Zhou*

1. **Introduction**

There are many benefits for students to take courses with someone they know. A friend in a course means a way to get notes for a missed class, another perspective on a difficult problem, someone to review with, and more. Some majors, such as computer science, require students to find partners or team members to work collaboratively throughout the semester. Students have been able to find partners or friends in a course by talking to them in person, but what’s lacking in this interaction is any information about the potential partner. With the prominence of apps like Tinder, students currently live in an environment where they are able to learn about someone before they choose to interact with them. Bindr will allow students who are mutually interested in finding a study partner to learn about their options before reaching out to them.

1. **Proposed System**

“Bindr” is an Android app intended for the use of students to make informed decisions on which classmate would be ideal for them to study and work with. This version of Bindr will be created specifically for students of Rutgers University, New Brunswick. Users will be able to create an account. The account will have the following information: A profile picture, courses (given by name and Course ID), a biography, and interests, and optionally a GPA. At its core, Bindr will provide users with a comprehensive list of potential students they can choose to match with. During this process, users will be able to make the decision of whether or not they want to study or partner up with them by evaluating the other user’s profile. The data they are evaluating on another user’s profile is a result of that user’s previous study sessions with other users. We are aiming to provide total transparency for students with Bindr by allowing users to rate the effectiveness of their recent experience with another student, whether it was a study session, homework session, or project assignment. Furthermore, the integrated messaging system on Bindr will allow users to start talking with one another the moment a match has been found, so they can get studying sooner. With all these features in mind, Binder aims to provide the ultimate solution to a problem that many students face.

* 1. *Functional Requirements*

**Input**

1. IN-001: User shall input whether they want to register an account or login to an existing account.
2. IN-002: User shall input username/email and password for login.
3. IN-003: User shall input username, email, password, full name, biography, interests, courses (name and Course ID), and optionally GPA for initial registration.
4. IN-004: User shall upload profile image for initial registration.
5. IN-005: User shall be able to delete their account.
6. IN-101: User shall be able to edit parts of their profile (profile image, password, courses, biography, email, GPA, interests).
7. IN-102: User shall be able to pick “match” or “pass” on another student.
8. IN-201: User shall be able to send messages to matched students in chat section.
9. IN-202: User shall be able to set up reminder (alert) at an inputted time for a Study Session.
10. IN-203: User shall be able to block a student they are matched with.
11. IN-301: User shall be able to rate the focus, engagement, productivity, and environment of a past Study Session (x/5 stars for each category).
12. IN-302: User shall be able to set account status (active/inactive).
13. IN-401: User shall be able to filter chats by user or course name.

**Task/Database**

1. TK-101: System shall show users all other non-matched and non-passed users for a given course in the user’s inputted courses
2. DB-101: System shall store user information (username, password, email, full name, biography, interests, GPA, courses) in a database.
3. DB-102: System shall record “match” or “pass” for each user on other users.
4. TK-201: System shall use Scaledrone to transfer messages.
5. TK-202: System shall not show students blocked by a user as potential matches.
6. DB-201: System shall store previous chat messages in a database
7. DB-202: System shall remove user information (username, password, email, full name, biography, interests, GPA, courses) from database when user deletes account.
8. TK-301: System shall send in-app reminder alert for Study Sessions
9. TK-302: System shall update user’s rating after Study Sessions as the average of the user’s focus, engagement, productivity, and environment ratings (as inputted in IN-301).
10. TK-401: System shall generate filtered list of matches (as inputted in IN-401).

**Output**

1. OT-101: System shall display Login Screen.
2. OT-102: System shall display Registration Screen.
3. OT-103: System shall display User Profile Screen including profile picture, name, courses, biography, interests, (optionally) GPA, and current average rating.
4. OT-104: System shall display list of courses to match in.
5. OT-105: System shall display un-matched and un-passed users one at a time to the User for a single course.
6. OT-201: Matched Users will be able to see the other user in their “Chat” section.
7. OT-202: System shall display messages in chat.
8. OT-301: System shall display active/inactive status for users in the “Chat” section.
9. OT-302: System shall send in-app reminder alerts for upcoming Study Session at the requested time of the alert.
10. OT-401: System shall display filtered results of matches.
    1. *Non-Functional Requirements*
11. User matches should appear within 3 seconds of the user opening the matches page of the app.
12. The system should operate as an Android application on an Android mobile phone.
13. The application should be usable only to students at Rutgers University.
14. The system shall support at most 20 users chatting concurrently.
15. The system should determine whether given email/username (during login) matches the password within 5 seconds.
16. The system shall display each student card within 5 seconds.
17. The system shall filter and displays matches within 2 seconds per match that Student has (e.g., if Student has 4 matches, Bindr must filter and display filtered matches within 8 seconds).
    1. *System Models*

Below is a comprehensive list of the use cases for Bindr. There are three subsystems: User Profile (the functionality to login, register an account, edit a profile, set account status, and delete account), Study Session (the functionality to set up a study session and rate a past study session), and Matches (the functionality to find potential matches, block students, chat with matched students, and filter matches).

* + 1. Use Cases

|  |  |
| --- | --- |
| *Use Case Name* | Login |
| *Participating Actors* | Initiated by Student |
| *Flow Of Events* | 1) Student chooses to log into Bindr.  2) Bindr prompts Student to input username or email, and the password  3) Student inputs either username or email and the password  4) If inputted email or username does not match the password, Bindr responds with “Incorrect username/email or password” and Student goes back to step 3. Otherwise, Bindr displays the Home. |
| *Entry condition* | * Student has app installed and is not logged in. |
| *Exit conditions* | * Student has successfully logged in, OR * Student chooses to not log in by closing app. |
| *Quality requirements* | * Bindr determines whether given email/username matches the password within 5 seconds. |

|  |  |
| --- | --- |
| *Use Case Name* | RegisterAccount |
| *Participating Actors* | Initiated by Student |
| *Flow Of Events* | 1) Student chooses to create an account on Bindr.  2) Bindr responds by presenting the sign-up form to Student  3) Student completes the form by inputting username, email, password, profile picture, biography, interests, school information, courses, and optionally GPA. Once the form is completed, Student submits the form.  4) If inputted email or username is already in use in another Bindr account, Bindr responds with “Email in use” and/or “Username in use” and Student goes back to step 3. Otherwise, Bindr displays “Account successfully created” and displays Student’s completed profile page. |
| *Entry condition* | * Student has app installed and is not logged in. |
| *Exit conditions* | * Student has successfully signed up, OR * Student chooses to cancel sign-up. |
| *Quality requirements* | * Bindr saves Student information. |

|  |  |
| --- | --- |
| *Use Case Name* | EditProfile |
| *Participating Actors* | Initiated by Student |
| *Flow Of Events* | 1) Student chooses to edit profile information.  2) Bindr responds by presenting the edit profile form to Student.  3) Student chooses to edit their profile picture, password, classes, GPA, school info, and/or interests. |
| *Entry condition* | * Student is logged in. |
| *Exit conditions* | * Student saves this information, OR * Student cancels the edit. |
| *Quality requirements* | * Bindr saves Student information. |

|  |  |
| --- | --- |
| *Use Case Name* | SetAccountStatus |
| *Participating Actors* | Initiated by Student |
| *Flow Of Events* | 1) Student chooses to set account status as “Active” or “Inactive” |
| *Entry condition* | * Student is logged in. |
| *Exit conditions* | * Student chooses to go back to Home. |
| *Quality requirements* | * Student’s status is viewable by other students after the other students refreshes their chats with Student. * Inactive students are not displayed as potential matches to others. * Other students cannot send messages to an inactive student (but can still see previous messages). |

|  |  |
| --- | --- |
| *Use Case Name* | DeleteAccount |
| *Participating Actors* | Initiated by Student |
| *Flow Of Events* | 1) Student chooses to delete their Bindr account.  2) Bindr prompts Student on whether they are sure want to delete the account  3) Student selects “yes” or “no”  4) If Student selects “yes”, Bindr displays “Account successfully deleted” and displays the login screen. |
| *Entry condition* | * Student is logged in. |
| *Exit conditions* | * Student receives an acknowledgement from Bindr that their account has been deleted, OR * Student cancels the delete. |
| *Quality requirements* | * After Student deletes his/her account, all the Student’s chat messages with other matched students will be removed. |

|  |  |
| --- | --- |
| *Use Case Name* | FindPotentialMatch |
| *Participating Actors* | Initiated by Student |
| *Flow Of Events* | 1) Student chooses to find potential matches.  2) Bindr responds by presenting a list of courses that Student has saved on their profile.  3) Student chooses a course for which to find potential matches.  4) Bindr displays the card of a student who has this selected course on their profile.  5) Student decides whether to “match” or “pass” on displayed student. Then go back to step 4. |
| *Entry condition* | * Student is logged in. * Student has courses saved in their profile. |
| *Exit conditions* | * Student chooses to stop finding matches, OR * There are no other students in the selected class who Student has not chosen to match with/pass on. |
| *Quality requirements* | * Bindr displays each student card within 5 seconds. * Bindr saves choice of “match” or “pass” for each student card. |

|  |  |
| --- | --- |
| *Use Case Name* | ChatWithMatchedStudent |
| *Participating Actors* | Initiated by StudentA  Communicates with StudentB |
| *Flow Of Events* | 1) StudentA chooses to chat with StudentB  2) Bindr responds by presenting previous messages (if any).  3) StudentA writes a message and sends it.  4) If StudentB is logged in, StudentB receives in-app notification about new message from StudentA. |
| *Entry condition* | * StudentA is logged in. * StudentA and StudentB are matched. |
| *Exit conditions* | * StudentA chooses to go back to Home |
| *Quality requirements* | * Bindr saves messages for both StudentA and StudentB regardless if StudentB is logged in or not. |

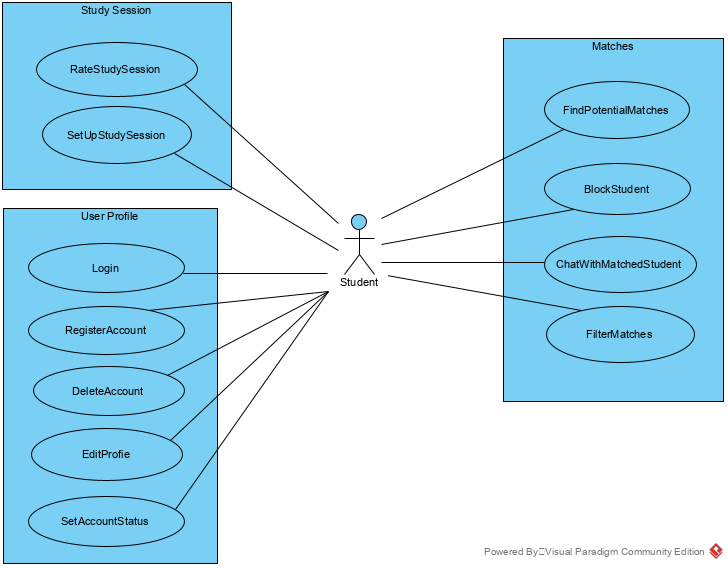
|  |  |
| --- | --- |
| *Use Case Name* | FilterMatches |
| *Participating Actors* | Initiated by Student |
| *Flow Of Events* | 1) Student chooses to filter by course name or course ID and/or username.  2) Bindr displays students who have matched with Student who meet the filter criteria. |
| *Entry condition* | * Student is logged in. |
| *Exit conditions* | * Student chooses to chat with a displayed student, OR * Student chooses to go back to Home. |
| *Quality requirements* | * Bindr filters and displays matches within 2 seconds per match that Student has (e.g., if Student has 4 matches, Bindr must filter and display filtered matches within 8 seconds). |

|  |  |
| --- | --- |
| *Use Case Name* | BlockStudent |
| *Participating Actors* | Initiated by Student |
| *Flow Of Events* | 1) Student chooses to block a student with whom Student is matched.  2) Bindr prompts Student on whether they are sure want to block the chosen student.  3) Student selects “yes” or “no”  4) If Student selects “yes”, Bindr removes the blocked student from Student’s matches and removes Student from the blocked student’s matches. |
| *Entry condition* | * Student is logged in. |
| *Exit conditions* | * Student receives an acknowledgement from Bindr that the chosen student was successfully blocked, OR * Student cancels the block |
| *Quality requirements* | * After Student blocks the chosen student, neither student can view the chat between them or the other student’s profile. * Neither student will be shown in potential match cards for the other student after the block. |

|  |  |
| --- | --- |
| *Use Case Name* | RateStudySession |
| *Participating Actors* | Initiated by Student |
| *Flow Of Events* | 1) Student chooses to rate a completed study session.  2) Bindr presents Student with the study session attributes “Focus”, “Productivity”, “Engagement”, and “Environment”  3) Student rates each of these attributes from 1 to 5. And then student submits or cancels these ratings. |
| *Entry condition* | * Student is logged in. * The study session was in the past. * The study session was not rated already. |
| *Exit conditions* | * Bindr acknowledges submitted ratings, OR * Student cancels ratings. |
| *Quality requirements* | * Bindr saves ratings. |

|  |  |
| --- | --- |
| *Use Case Name* | SetUpStudySession |
| *Participating Actors* | Initiated by StudentA  Communicates with StudentB |
| *Flow Of Events* | 1) StudentA chooses to set up a study session with StudentB.  2) Bindr prompts StudentA to select a start time.  3) StudentA selects time and picks whether/when to get an alert for this study session, and then StudentA submits request.  4) Bindr sends this request as a message to StudentB in the chat between StudentA and StudentB, with the option for StudentB to accept or decline.  5) StudentB accepts or declines request.  6) If StudentB accepts, Bindr prompts StudentB to choose whether/when to get an alert.  7) Bindr alerts StudentA of StudentB’s decision |
| *Entry condition* | * StudentA is logged in. * StudentA and StudentB are matched. |
| *Exit conditions* | * Bindr acknowledges study session, OR * StudentA cancels study session setup. |
| *Quality requirements* | * Bindr saves study session hours. |

* + 1. Use Case Diagrams

**

1. **Glossary**

|  |  |
| --- | --- |
| **Term** | **Definition** |
| Bindr | The name of the system. |
| Course | A class offered on the Rutgers University academic catalog. |
| Course ID | A three part number representing a course at Rutgers University. Example: “01:198:111” for Introduction to Computer Science. |
| DB | Database functional requirement. |
| Home | The Home screen. Also the courses match page. Shows the list of courses the User has inputted. |
| IN | Input functional requirement. |
| Match (Noun) | An agreement between two users where both agree to chat with the other user. |
| Match (Verb) | A single User’s choice indicating they would like to chat with another specific user. |
| Matched Users | A pair of Users that are in a Match (Noun). |
| OT | Output functional requirement. |
| Pass | The opposite of Match (Verb). A single user’s choice indicating they would not like to chat with another specific user. |
| Scaledrone | The backend chat API that we will use to power our chat feature. |
| Student | Someone currently enrolled in at least one course at a University. The intended User of this system. |
| Study Partner | For a given Student, a Study Partner is one of the student’s classmates who will study or work on collaborate assignments with them. |
| Study Session | A time when matched users have agreed to meet in person. |
| Tinder | A dating app that describes itself as “the world’s most popular app for meeting new people”. Functions by allowing users to swipe left or right on potential dating partners. Two people who both swipe right on each other are allowed to chat with each other. |
| TK | Task or process functional requirement. |
| UI | User Interface function requirement. |
| User | Someone who interacts with the Android application. Also a Student. |

1. **Reference**

[**https://www.scaledrone.com/blog/android-chat-tutorial/**](https://www.scaledrone.com/blog/android-chat-tutorial/)

[**https://tinder.com/**](https://tinder.com/)

[**https://www.android.com/**](https://www.android.com/)