Deliverable: Product Evaluation Document

Part 1: Testing

Internal Testing

Throughout the development process, all new methods created were unit tested on local development branches and then merged to the main branch. If any new bugs were detected by any member of the team, the issue was immediately reported, collected on a shared note, and cleared by a member. In this way, the development process included thorough internal testing.

Checking for function/method failures: (Checking function return values. Considering exceptions thrown by functions/methods.)
Our database API functions raise specific errors in the cases of invalid parameters, or broken invariants. For instance, there are many requirements that need to be satisfied when making a new request—the request must contain at least one time; both users must be available at the times; both users must be open, etc. Each invalidity raises a unique error, which is propagated to and handled by the Flask application error handlers registered for each error, and a response is generated accordingly for the client.

Validating parameters
As we note in the previous section, our database API function calls validate that parameters are valid, particularly for request operations. For non-primitive parameters, such as lists and dictionaries, we perform a limited amount of validation—for instance, whenever we work with a schedule list, we assert that the schedule is of the correct length.

Checking invariants
For example, schedules arrays are always validated using an assert statement within db.py (schedule_to_events()) before they are converted into a string representation.

Leaving internal testing code intact
Asserts were used and left in the database files - specifically db.py, matchmaker.py, request.py, schedule.py, and user.py. Request.py and user.py included asserts checking whether the current session was active before accessing any session variables.

External Testing

White box external testing:
Boundary (corner case) testing:
There are three main inputs from the user.

The completed fields for the profile (name, contact, bio) were tested for strange inputs (including no input, long input, invalid formats, special characters, etc). These corner cases were accounted for through a custom input validation function written in jQuery. The corner case of not indicating a gender preference is covered in this validation function as well (will prompt users to "choose at least one field"). However, after testing it was found that the system takes empty "Interests" fields (all checkboxes unchecked). For the purposes of the application and matchmaking, this should not be possible and should have been validated.

The other main input is the selection of times from the Find a Buddy calendar, Incoming Requests calendar, and the Profile availability. Users are not able to select invalid times on the Profile calendar. They are also not allowed to select times beyond the gray boxes for the Find a Buddy Calendar and the Incoming Requests Calendar, as the UI renders the other boxes unclickable.

Statement(coverage) testing:
We had a couple of strategies to perform coverage testing. With the Find a Buddy UI, it can sometimes be hard to achieve specific kinds of behavior in the application; instead, we sometimes used the Master Debugger to test specific users and requests. To check if certain statements are executed, we had the server print function parameters and other information at numerous points, and then analyzed the output log of the server for a given series of inputs.

Path testing:
Complete path testing was simply unfeasible given the complexity of the web application. However, as many user routes as possible were tested, and appropriate popups were displayed to guide users to the correct input or next action.

Black box external testing:
Use case testing:

We followed the user's guide to complete the tasks with strange inputs. The same flaws were detected for the profile "Interests" fields (allowing them to be empty) and the empty Profile Availability calendar.

Regarding Incoming Request calendar and Find a Buddy inputs, even if a user were to alter the HTML and select times not grayed, the system correctly rejects the times by displaying an error message. Both calendars only accept valid requests with times selected (no empty times). If a user receives an Incoming Request and clicks "Modify Request" without changing times, they will be correctly prompted to "Please change at least one time". However, if they select a new time but click "Accept", the system does not detect the change and intention of the user to modify the times. Instead it just accepts the original request, ignoring the new user's changes. It would be better for the UI to hide the "Accept" button if the user makes any change to the times.

There is no validation for the input for blocked users. The user is able to enter any form of text, including special characters, spaces, etc. despite the fact that the search box is for netids. Invalid text has no effect on the backend (which is the correct behavior), but for the user the system will not give any message stating why the input is invalid, instead just showing no change in the blocked table. This could be a problem if a user simply mistakenly put a space after an existing netid, but the system does not block the user.

XXS attacks were attempted at profile text fields. Profile fields protect against these by taking the XXS attack as a literal string thanks to string sanitation by Jinja2. The app uses SQLAlchemy, which protects against SQL injection attacks. CSRF attacks are protected against attaching CSRF tokens to every POST request. With this, it is also notable that we are confident in our database protection. This is because the queries to the database are restricted to CAS-Authenticated users. Queries are properly handled/sanitized/validated from the front-end before they are passed to the database.

Stress testing:
For example, we have tried the following method of testing how many different active requests a person can have. Logged in as netids jasono and ejcho, we went to Find a Buddy and sent 20 different outgoing requests to people and verified that all the match requests were sent properly. To simulate many simultaneous connections to the server, we had 20 tabs open in an attempt to overload the server with requests.

Field external testing
Please see the user evaluation section below.

Unresolved issues
Our database connection is one of the more fragile parts of our product. All pages within our application are dynamically updated through Ajax and jQuery, and every page sends requests every second to the server. For each request, the client attaches a timestamp for the last time that they refreshed; the server uses the timestamp of the user to determine whether or not a refresh is necessary. This allows us to perform live updates without having to query the database for all of the information every time. But these constant queries to the database can occasionally fail—for instance, there are connection issues when a user has a page open for too long, and the error message - “Oops! An unexpected error occurred. Please refresh” displays. We speculate that this is due to the many connection pools.

Known bugs/flaws:

1. Malicious actors who are CAS-authenticated could change the request id in their post requests to alter requests that belong to other users. The database does not validate request ownership before performing a transaction.
2. Logging out and then

3. Our app was built for the web and is not compatible with mobile devices.
4. Our app was built for and tested only on standard browsers (Chrome, Safari).
5. The calendar UI does not allow for mass deselection of times using drag features.
6. The profile allows for empty “Interests” and “Availability”. As “Availability” is a hard filer on the matchmaker finding potential partners, this is a pretty big flaw. The system will not alert users that the reason they are not seeing candidates is due to an empty schedule. This is a bug that was discovered too late and thus not addressed.
7. There is no validation for the input for blocked users. The user is able to enter any form of text, including special characters, spaces, etc. despite the fact that the search box is for netids. The system will not give any message stating why the input is invalid, but will instead just show no change in the blocked table.
8. Users are allowed to press “Accept” in the Incoming Requests module even after altering the times, which will accept the original times proposed to finalize the match. The system does not understand the intention of the user to modify the times.

Robustness

In terms of robustness, many parts of our application are well-protected from errors and bad inputs, and work well as expected. As mentioned previously, our user profile is robustly validated and protected from any user input that future functions or our database wouldn’t accept–for example, we validate that the phone number that is input is a 10-digit number, as we need this information to send SMS messages correctly. We also warn the user whenever a profile input is faulty or not in the format we require, which allows them to make changes. As also explained above in the black box external use case testing, we tested to make sure our database was protected from any SQL injections, XXS attacks, etc. This is extremely important since whatever the user inputs is saved into the database. Thus, we ensure that no user input to our database is faulty or bad. We also included backwards-compatibility to ensure that every time we made changes to the profile inputs, we required users with old profiles to update their profiles before being able to use the application again. This meant that we could make sure that everyone’s profiles, and therefore the database columns, are as they should be.

In addition, we checked many corner cases that could appear while users send requests to each other, making the processes of sending and receiving requests robust by having pop up errors to protect against unwanted or breaking changes. For example, when a user attempts to confirm a request with a buddy, if there are any active requests between the user and others with conflicting times, a warning popup will appear to allow the user to confirm whether they want to continue and cancel all other requests. In addition, error messages that deal with live updating also warn users whenever they are viewing information that is changed as they are looking at it. For example, if a user opens an incoming request as the sender blocks that user, a message will show up informing the user that the other buddy is no longer available for matching. In addition, the system ensures that multiple active requests cannot be sent between the same two people, and

will throw an error popup in that situation as well to prevent any unwanted actions to occur. In this way, our request features are entirely robust and protected by error handling.

Our Calendar UI and HTML is robust as well–as explained above in the black box use case testing, altering the HTML of the Find A Buddy page calendar UI is protected against and throws an error, having no effect on the requests that a user sends to another. Every POST request is also protected from CSRF attacks, meaning that any information the user sends to the server such as after creating the profile or sending a request is protected as well from malicious attacks. Due to use of SQLAlchemy and Jinja2, we find that our database and any data passed to and from the database is robust and well-protected.

Part 2: User Evaluation

We conducted group and individual evaluation sessions. The group sessions were helpful in the sense that the testers could send and accept requests to each other and thus had a heightened sense of "interacting" with real users on the platform. The complete task list and compiled notes across the sessions can be viewed in the appendix. Quotes were written down to keep a more accurate record of the sentiments and nuances behind every feedback.

Generally, the users were able to complete the tasks well. In particular, they were easily able to create the profile and enter valid input into the fields, guided by the feedback provided (if invalid, a red outline and warning sign with a message saying why the input is invalid shows up. If valid, a green outline). The majority of users claimed that the fields in the profile made intuitive sense as to why they were important - "It's pretty clear and all the fields made sense." In particular, some stated that the additional information section was helpful and provided an extra space for them to describe anything that another user needed to know about them before matching. Users had more varying opinions about the interests field - the majority said that the current selection of interests was broad yet specific in the sense that they encapsulate goals people could have at Stephen's fitness center within Dillon, and that the intention of the app to be used to find people to go to Stephen's fitness center was clear. However, some users expressed some disappointment, saying "I'd also like someone just to play basketball with at Dillon," perhaps because they have a broader definition of the gym rather than just lifting weights. For the availability portion of the profile, some users needed a bit more time to figure out the drag feature, which is slightly different from that of other calendar UIs (ours requires a single click and then hovering over other blocks to mass-select them - other UIs like Google Calendar require the user to keep their mouse clicked as they drag). However, after one or two tries, all users were able to understand the function and select times with ease. Many were intrigued by the calendar UI's ability to "merge" time blocks - if two time blocks were selected right next to each other, the calendar merges the time blocks into one large one. Users did comment on the lack of an undo button for deselecting times, as they would have to either manually click every

time slot to deselect it or clear the entire selection and start over. In addition, some stated that the time blocks are not clear because there are no gridlines. However, they did say that the little text at the start of the block showing the time range of the entire selected time block cleared the confusion.

For the Find a Buddy page, people generally seemed confused at first. Although there was text describing what actions to take, some people did not read the text and immediately tried the buttons “Match” or “Pass” below. For these people, the facilitator of the feedback session would direct them to read the instructions at the top of the page, which told them how to match or pass. At this point, they were able to get a clear understanding of what to do and continued with ease. Those who read the tutorial were able to understand the page easier. Another source of confusion was the “No more matches” message that occurred when users either went through their list of candidates provided by the matchmaking algorithm or if their profile was so restricted to not have any overlap with other people’s profiles. They said that they were a bit taken aback by the message because it seemed like they “only had like 2 candidates and then ran out”. Perhaps changing the message or making the actions items clearer would prevent this confusion. Regarding the scenario where a user was viewing a possible candidate, they commented that it would be nicer if the “Match” and “Pass” were immediately visible, but unfortunately our calendar UI implementation is a bit large and could not be reduced. Thus, some users with smaller screens have to scroll to see the buttons, which is inconvenient. Overall, people said that they expected a process of finding a buddy first and then coordinating schedules. This is why some users kept forgetting to suggest times before pressing “Match.” After learning that the app allows for both simultaneously (searching for a buddy based on availability), they were very pleasantly surprised because grouping the tasks together reduces the expected amount of work - “Ohhhh, so I can see their profile and availability. I’m not just matching based on profile and then scheduling after? That’s nice.”

For the incoming requests page, users commented a lot more on the modifying request feature than the accept or delete request. They were able to accept and delete with ease since the function is very clear. Regarding the modification, some users stated that it was confusing to see both the “Accept” and “Modify” button together when you first open up the request, stating “It would be nice if the ‘modify’ button only appeared after you actually modify.. If I just open up the request and see both buttons initially it’s confusing”. However, regarding the function itself, users said that it was probably the “most helpful feature [...] it gives me more flexibility.” They liked in particular that you could “continue coordinating with someone you really like instead of having to delete the request because you don’t like the times.” Initially, users were a little confused about the back and forth nature of modifying incoming requests (essentially, that you could continue to send modified requests to each other). However, after experiencing a couple modified requests, they stated that it was “kinda fun to keep ping ponging requests” and that this “really sets the match in stone since both people have a say in the final match times”.

The outgoing requests and matches page was very self-explanatory to users, they didn’t have too many comments on the page. They said that it was nice to be able to keep track of their

current outgoing requests and delete them if necessary. Some expressed disappointment that they could not modify outgoing requests. Users enjoyed the fact that they could unmatch with people later on if they did not find them to be good gym partners.

The settings page was also fairly intuitive for users. They enjoyed the fact that you could enable SMS notifications, saying “Love how you have SMS notifications… I don’t want to login to the app every time to check!” They also expressed contentment with the fact that we implemented blocking, saying that they definitely might have people they do not want requests from or that this would prevent spamming. However, they said that it would have been more convenient if they could block someone they encountered on Find a Buddy. Currently, this is not possible because the profiles in Find a Buddy do not show netid, which is needed to block someone. Finally, some users commented that the “Delete Account” functionality was useful because they “don’t want [their] number [...] floating around online.”

Users also made general comments about the UI and design choices. For example, they enjoyed the motivational message “You look great today” on the Dashboard page when they first login, saying that it really “aligns with your goal of promoting fitness” in a positive way. They said it would be better if there was a rotation of such messages. Regarding the general design (color scheme, layout, etc), all users said it was very “clean,” the color scheme is “appropriate,” and the “navigation bar is nice.”

To summarize, users appreciated the concept of matching with gym buddies based on both availability and interests. They found the modifying requests function especially helpful in solidifying matches and ensuring that both parties were satisfied with the times. Users said they like the notification system (mostly the SMS) because it was convenient for them to receive updates through text rather than having to login. They said overall the concept of matching was fairly clear, except for some details such as the “Match” and “Pass” in Find a Buddy. Even this small confusion was quickly resolved after one or two attempts. Regarding the question of whether they would use the app, the majority of users said that they would. For example, one user who was new to the gym said, “Never worked out before but always wanted to get into it. Just never knew how and never had anyone to do it with.” Experienced gym-goers also said that this app would satisfy their needs, saying “I really need a partner who will spot me because none of my friends consistently go. I think this app will allow me to find one.” Another person claimed that this app could be used for other purposes, saying “Really cool app. I’d use it not only to find a new gym partner but also just to make friends.” On the other hand, not all users were excited. One user thought the process was too intensive, saying “I'm not sure if I would use this… It seems like a lot of work to find a gym partner.”

Part 3: Expert Evaluation

It must consist of your own formal evaluation of your product using the heuristic evaluation technique. Optionally, you also should evaluate some important parts of your product using the

cognitive walkthrough technique. Both heuristic evaluation and cognitive walkthrough are described in lectures.

Heuristic Evaluation:
(1) Visibility of system status

- The system should always keep users informed about what is going on, through appropriate feedback within reasonable time.

Our system gives appropriate feedback to users through waiting cursors and pop ups. For example, after the user has clicked “Match” on the Find a Buddy page, the pop up saying “Request successfully sent!” will display to assure the user that their request has been sent. Immediately after the button is clicked and before the popup loads, the cursor will change to a waiting cursor to signal that the app is processing the action. Several other times the cursor changes to a wait cursor is after accepting/modifying/deleting incoming requests, deleting outgoing requests, clicking “Pass” on Find a Buddy, refreshing the Find a Buddy candidate list, clicking “Update” on Profile, clicking “Unmatch”, clicking “Block”, and clicking “Delete Account.” These wait cursors are followed by a message stating that the action has been completed for more important actions, such as “Match has successfully been finalized.”

In addition, there are descriptions on specific pages that instruct the user on how to use parts of the application that might lead to some initial confusion, such as the one instructing how to operate the calendar UI on the Find A Buddy page, which reads: “To match: select times out of the gray boxes and click "Match" to request them. Selected times will appear in blue.” Other descriptions such as the one below the Phone Number field on the Profile page inform the user to only put down a U.S. SMS phone number, since that is the way through which they can be sent SMS notifications about updates.

In order to inform users about new changes, we enabled both in-app and SMS notifications. In-app notifications are sent when a user has a new incoming request or a match has been finalized (i.e. their outgoing request has been accepted by someone). SMS notifications are sent when a user has a new incoming request, a user has a new modified incoming request, a match has been finalized, or a match has been canceled. The SMS notifications are a bit more descriptive (include name AND netid) as well as cover more cases because the user is assumed to not be on the app and thus needs more updates. Both in-app and SMS notifications are sent immediately.

When the system encounters an error, an error message is immediately displayed to inform users of the error and its solution. More details about the error messages are below.

(2) Match between system and the real world

- The system should speak the user's language, with words, phrases and concepts familiar to the user, rather than system-oriented terms. Follow real-world conventions, making information appear in a natural and logical order.

The language used in the system consists of familiar terms such as “request”, “match”, “incoming”, “outgoing”, “buddy”, and so on. These words should be common to users and especially the target audience of Princeton students. In particular, students of this age are likely familiar with the concept of passing or matching with people due to widespread usage of popular matching apps such as Tinder. In addition, the concept of sending modified requests back and forth is intuitive and students can be found doing this in their daily lives when they are coordinating times to meet others. The use of the term “buddy” also has a familiar tone, which helps put users at ease when matching with others as they know it is for friendly purposes. The drag feature of the calendar UI is fairly intuitive, as many students use Google calendar or other drag calendar UIs for their daily schedule.

Information is displayed in a natural and logical order. For example, on the Find a Buddy page the user first sees the person’s profile and availability before they encounter the “Match” and “Pass” button, since a user would want to read and evaluate the potential candidate before taking action. The navigation bar is ordered based on logical importance - “Find a Buddy”, “Incoming Requests”, “Outgoing Requests”, “Matches”, “Profile”, “Settings”. Note that this order also corresponds to the amount of action that needs to be taken - people must make the most decisions in who to match with in Find a Buddy, decide whether to accept/modify/delete incoming requests, look at outgoing requests and potentially delete one, or just view matches (less likely to unmatch).

Currently, one part that could potentially be confusing for users is the message “No more matches :(” on the Find a Buddy page. This message appears in one of two cases: 1) when the users have passed through their current list of candidates, or 2) when the user has no overlaps in schedules or no matching profiles with anyone else on the app. It would be useful for our app to clearly distinguish between the two cases so that users are not left wondering of the reason why they have “no more matches.” This phrase could potentially be unclear about whether there are just no more people available for them to match with or if their profile just needs some tweaking.

(3) User control and freedom

- Users often choose system functions by mistake and will need a clearly marked "emergency exit" to leave the unwanted state without having to go through an extended dialogue. Support undo and redo.

User control and freedom is centered around their inputs to the application. The biggest “mistakes” that people could make are with the calendar UI. In particular, when filing out their availability calendar users can select large blocks of time within the same day easily using our drag feature. With the current calendar UI, users unfortunately cannot deselect using drag - they must manually click every selected time to deselect it. In case a user selected too many times accidentally, there is a “clear” button, which clears the entire selection of times. We decided that having an “undo” button would be too complicated, as the system would have to keep track of

the history of the time blocks added in order to undo them in order. However, this would be a convenient feature to add, if we had enough time.

To address this issue of selecting too many times and then having to manually deselect them, in the Find a Buddy page and Incoming Request modification pop up, the calendar UI does not allow for drag selection of times. Since people are likely only going to be working out for an hour or two a day, they would only need to select a couple time blocks overall, and the mass-selecting drag functionality was deemed unnecessary. It would be easier to click on one time slot and just unclick the time slot to deselect it.

Other mistakes that users can make are accidentally sending an outgoing request to someone from Find a Buddy. In this case, they can delete the request by navigating to the Outgoing Requests tab and clicking the "Delete" button.

Some mistakes that are not easily reversible (but should have been given enough time) are the deletion of an incoming request and accepting an incoming request. In addition, there is also the case where a user may have misclicked and selected an extra time to propose to a user. Since no outgoing request modifications are enabled, this change cannot be easily reversible, though it is a very plausible error. To further improve the system while minimizing confusion on the side of the person receiving the request, we could allow for outgoing request modifications for a short while (1 minute) after the request is sent.

(4) Consistency and standards

- Users should not have to wonder whether different words, situations, or actions mean the same thing. Follow platform conventions

All language is standardized across the pages. For example, the term "request" is used to describe two potential partners who are coordinating times, and the term "outgoing" and "incoming" is used to indicate the direction of the request. The term "Match" is used in the Find a Buddy page to suggest times to a potential gym partner, and the same term "Matches" is used to describe the finalized gym partnerships. In order to get rid of an incoming or outgoing request, the same term "Delete" is used.

The UI is kept consistent as well. For example, the same calendar is displayed consistently across pages where times need to be selected, such as the Profile calendar of availability and the request calendar for Find a Buddy and Incoming Request modifications. In addition, the profile cards for the Profile page and the New user profile creation page are the same (more in step 6).

(5) Error prevention

- Even better than good error messages is a careful design which prevents a problem from occurring in the first place. Either eliminate error-prone conditions or check for them and present users with a confirmation option before they commit to the action.

We have several rounds of error prevention. Let's consider the task of profile creation. From the front-end, validation for inputs is done using a custom jQuery validation function that prevents the submission of invalid inputs from the profile creation. In case the information makes its way to the database, the database does a check on profile validity every time a user is logged in. It will redirect the user to the profile editing page and force them to create a valid profile. This also enables backward compatibility, enabling developers to make changes to the profile (e.g. include more fields) without allowing existing users to have an invalid profile by making sure to prompt them to fix it before they can use the rest of the app.

There are pop ups notifying the users of issues with the Find a Buddy page. The profile includes an "open" button to allow users to set their availability to match with others. Toggling on would allow them to send and receive requests, and toggling off would hide their profiles from others and they would stop receiving requests. An example for such a case would be when they have just finalized a match and are not looking to find a new partner as they test out their match. For new users who are joining the app, the assumption is that they want to immediately find a gym partner; thus, the default for the toggle is 'on.' In case users toggle it off accidentally and navigate to the Find a Buddy page, they will encounter a message saying "Oops! Please toggle Match Availability in Profile to Open in order to enter Find a Buddy." Additionally, since the Find a Buddy page is a place for users to submit workout times to a potential gym partner, a user cannot send empty requests. To account for the case where a user forgets to select times and clicks "Match," a popup appears prompting the user to select at least one time before submitting a request: "Please select at least one time."

Several confirmation popups are present before the user makes any important decisions including unmatching with a user, blocking a user, accepting a match, and deleting their account. To further explain the confirmation process for accepting a match, a user is only allowed to have one match per time slot. Thus, if they accept a match for a certain time (e.g. 9-10AM Saturdays), all other active requests (both incoming and outgoing) with the time 9-10AM Saturdays will be automatically canceled. We notify users of this through the following message: "Warning: accepting this request will cancel pending requests with ___'" and require them to click a button to finalize their decision. For blocking a user, accepting a match, and deleting the account, the confirmation popup restates the action they are about to take and requires another button to finalize the decision. Deleting accounts have a longer message clearly stating the consequences: "Are you sure you want to delete your account? By deleting your account, you won't be able to access Gymbuddies services. All your account information will be removed from the database. **This action cannot be undone**."

Currently, the system allows for a person to submit an empty availability schedule for their profile. This leads the user to having no possible gym partner candidates. When they navigate to the Find a Buddy tab, they see the message "No More Matches" followed by two buttons of "Refresh Matches" or "Edit Matching Preferences." The intention behind the button "Edit Matching Preferences" is to prompt users to edit their profile in such a way that they can find partners who fit them. However, since our app doesn't throw an invalid profile error for

empty schedules or display a corresponding message for it, the user may not be aware of the reason why they are not seeing any potential candidates–this is another problem we could fix with more time.

(6) Recognition rather than recall

- Minimize the user's memory load by making objects, actions, and options visible. The user should not have to remember information from one part of the dialogue to another. Instructions for use of the system should be visible or easily retrievable whenever appropriate

One example of recognition rather than recall is evident through our descriptive navigation bar: “Dashboard, Find a Buddy, Incoming Requests, Outgoing Requests, Matches, Profile, Settings” so that users do not have to wonder what each page contains. Moreover, all the buttons are given short and concise names describing the function (e.g. “Match” to match with a user, or “Pass” to see another user on the Find a Buddy page). Another concrete example is making times easier to view for users by displaying a draggable calendar UI when selecting times. This enables users to visualize the selected times better. However, for pages such as the Incoming/Outgoing Requests and Matches, we opted for a succinct summarized list of times. Since these pages display a list of several requests or several matches, it would be quicker to view times in written form rather than a large calendar per request.

For more complicated tasks, instructions are always included. The profile page has short descriptions of each field and placeholders indicating the format and examples of acceptable input (e.g. 1112223333 for the “Contact” field). The Find a Buddy page has a short description of what “Pass” and “Match” means, explaining that the gray boxes are time options to choose from and that the request should be submitted by clicking “Match.” Similarly, the Incoming Requests page (inside the pop up to view a request) includes descriptions on how to modify or accept a request using the “Modify Request” and “Accept” buttons. The actions involved in modifying a request from Incoming Requests are essentially the same as those in sending a match request from Find a Buddy. Thus, the layout of the Incoming Request modifying pop up(placement of calendar, gray boxes, profile on the right) is identical to Find a Buddy in order to trigger the user’s memory of how to send requests.

Similarly, the New User Profile Creation page looks almost identical to the Profile page to allow users to easily recognize what each field requires rather than having to read the short descriptions above the fields. In addition, the most important information about each gym buddy that a user matches with is displayed on important pages–for example, each incoming, outgoing, and match request displays essential information about a buddy that is visible to the viewer, such as contact information, requested times, etc, so that the user doesn’t have to constantly remember their information.

(7) Flexibility and efficiency of use

- Accelerators—unseen by the novice user—may often speed up the interaction for the expert user such that the system can cater to both inexperienced and experienced users. Allow users to tailor frequent actions.

There are several frequent actions that are highlighted by the UI. Once a user signs up for the first time and creates a profile, every time they log into the application afterwards they are taken directly to the dashboard without having to go through the profile page again, streamlining the process and allowing experienced users to get directly to viewing their matches and matching with others. In addition, once users finish using the calendar UI for the first time, they will be able to use the calendar more easily and quickly in successive times due to the intuitive drag-and-drop nature. In addition, the navigation bar is ordered in accordance with the most important functions. Since the app is for finding gym partners, the first link is Find a Buddy, followed by Incoming and Outgoing Requests, and then Matches. The Matches page is likely the least important page because only those with finalized matches would frequent it, if at all. Below these pages are the account pages. The profile precedes the settings page because it is likely that users will need to alter their profile more than needing to access account settings.

On the top navigation bar, there are three important icons. One is the profile icon, which also has the netid of the current user - this is convenient for the user to be able to know what account they are logged into at a quick glance. The next one is the notification bell, which turns red when there is a new notification. This is frequently used, so it is placed at a very visible spot at the top of the page. At the end, there is a logout button. This placement corresponds to the logout buttons of many other pages, and is thus a familiar spot where users will be looking for a quick way to logout.

In terms of other ways we increase efficiency of use, we highlight buttons that are more desirable than others. For example, since the goal is to match with users, from the Find a Buddy page we have the “Match” button as a dark blue that is more visible than the “Pass” button, which is white. Similarly, for viewing incoming requests the “Accept” button is dark blue and the “Modify Request” is white.

(8) Aesthetic and minimalist design

- Dialogues should not contain information which is irrelevant or rarely needed. Every extra unit of information in a dialogue competes with the relevant units of information and diminishes their relative visibility.

The UI is fairly clean in the sense that all text only includes information that is necessary. Descriptions are short and succinct, dialogues clearly explain the problem (in the case of error popups) and certain important actions are confirmed with pop ups as well. For example, once someone clicks “Match” in the Find a Buddy page, a popup saying “Request successfully sent!” shows. This message is concise and clearly confirms the action that was taken. The in-app and

SMS notifications are succinct as well. In-app notifications include “New incoming request from _netid_” and “Match finalized with _netid_”. SMS notifications are similar but include a short indication that the message is from GymBuddies. For example, a new incoming request SMS notification is “Hello from Gymbuddies. You have received a new match request from _name_ (_netid_)!”. Given that the user is not on the app and cannot just quickly navigate to the incoming requests page, the SMS notification is a bit more descriptive by including the full name of the requester.

In order to minimize the amount of words on the page, icons were used in the top navigation bar. A small human icon represents the profile, a bell represents the notifications, and a small arrow out of a box represents the logout button. These visuals make it easier for users to navigate without having to read extra words.

Overall, the overall color scheme and design of the pages is simple and minimalistic. We have a simple blue and white background with no flashy patterns diverting attention. All information is segmented into little cards for easy visual distinction. Buttons are clearly identifiable using a bold blue color contrasting a white background. When the window is small, the navigation bar is condensed into a little 3-bar icon (which can be clicked to expand it again) rather than covering important parts of the current page. None of the pages involve any substantial reading.

The exception to that is the tutorial page, which is a bit wordy. See step 10 for additional comments.

(9) Help users recognize, diagnose, and recover from errors

- Error messages should be expressed in plain language (no codes), precisely indicate the problem, and constructively suggest a solution.

We have several different error messages. Some are more specific, such as the ones mentioned in step 5 (error prevention). To recap them, we have pop ups to prompt users to select at least one time if they try to submit a request with no times, toggle the “open” if they have it off and try to access the “Find a Buddy” page, and display specific feedback about the format of inputs to the profile fields if they attempt to input incorrectly (e.g. input field turning red and showing the message “Phone number should have 10 digits, no special characters”). Another example of a specific error message is the case where a user 1 opens an incoming request, but in that instance the user 2 who sent the request to user 1 blocks them. In this corner case, if user 1 tries to accept or modify the match, they will get a message saying "The requested user is no longer available for matching” rather than receiving a generic error message. More examples include 1) the case for trying to create more than one request between the same users: “There is already an active request between __insert name__ and you,” 2) having someone cancel their request to you while you are viewing it: “The selected request no longer exists. Please refresh,” 3) having someone delete their profile while you are looking at their profile in “Find a Buddy”: “The requested user no longer exists. Please refresh,” 4) having someone (or even you) change

available times while you are looking at their profile in “Find a Buddy”: “Your match’s or your schedules have changed. The times you have selected are no longer available. Please refresh,” or 5) having 2 tabs open and logging out in one: “You've been logged out. Please refresh.”

Generic error messages arise when there is a backend server error. One generic error message is “The database is under load. Please refresh,” which occurs when SQLAlchemy throws too many operational errors. These errors can occur randomly, and are generally outside of the client and application's control. Thus, the solution is to just refresh and continue normal function. Another generic message covers all other unexpected errors. The message says “Oops! An unexpected error occurred. Please refresh,” prompting the user to just refresh and continue.

(10) Help and documentation

- Even though it is better if the system can be used without documentation, it may be necessary to provide help and documentation. Any such information should be easy to search, focused on the user's task, list concrete steps to be carried out, and not be too large.

Though we provide directions where necessary, some parts of our app may be less intuitive for users. For this reason, we take the users through a very short tutorial when they first create an account. In addition, this tutorial is always available on the side navbar through a button labeled “Tutorial” under the larger text “Need help? Please check our tutorial,” which goes through the basics of matching and acting on requests. We have also created a thorough User’s Guide with concrete use cases and specific instructions on how to achieve each function. To improve on the existing tutorial, it could be made interactive rather than a single scroll-through page, since visuals are a better explanation than words.

Appendix:

User Evaluation Sequence:

**** Little to no guidance was given to users during the task sequence in order to ensure that the app was self-sufficient in explaining.**

Brief Introduction:
For many people, going to the gym for the first time can be a confusing and intimidating experience. It can also be difficult for students to make time to work out, especially with their rigorous academic and social lives. For more experienced students, they may want to find partners with similar interests and routines that would push them harder to achieve their goals. Thus, there is a need for a way to allow Princeton students to find workout partners.

GymBuddies is a web app that provides a platform for finding a gym partner with similar interests and availability. It allows users to schedule regular workouts with their partner. In doing so, GymBuddies helps to promote exercise, which is an integral part of maintaining a healthy lifestyle at Princeton.

Task List

Part 1:

1. Sign up and create a new user profile.
2. Search for a gym partner in the Find a Buddy tab and propose times to go to the gym together.
    a. Feel free to pass users as desired.
3. View your outgoing request and observe what happens when they accept your request.
4. Navigate to the Matches tab to see your new match.
    a. Find a calendar showing your finalized matches.
5. Pretend you had a terrible experience with them, and proceed to block that person.

Part 2:

6. I will send you 3 requests from 3 different accounts. View the incoming requests. Decide what you want to do to each one - but you must do something different for each request.
7. Go to outgoing requests to see the modified request you made.
8. Go to Matches again (you should have 2 completed matches now).

Part 3:

9. Edit your profile and discover how this change affects the people you can match with in Find a Buddy.
    a. Tips: try to make your profile very restrictive if it was not and vice versa.

User Evaluation Notes:

**** The following are notes and quotes compiled across several user evaluation sessions. Repeated comments were not necessarily written down as separate quotes. ****

Overall sentiments:

- “Never worked out before but always wanted to get into it. Just never knew how and never had anyone to do it with. Now I think I can using this app”
- “I really need a partner who will spot me because none of my friends consistently go. I think this app will allow me to find one.”
- “Really cool app. I’d use it not only to find a new gym partner but also just to make friends”
- “I'm not sure if I would use this.. It seems like a lot of work to find a gym partner”

General/repeated notes:

- Understanding back and forth nature of request modifications requires a bit more time
- Some people repeatedly forgot to select times after pressing match
    - Although there is a written description that states that you need to select times, they don't read it and the natural instinct is to just click match
    - Improve UI so they remember and are spurred to select times first
- When they encountered the "no more matches" page after clicking pass on their selected candidates, they didn't know what to do next or seem to understand why there was a "edit matching preferences" option.
- When people were asked to change their matching preferences, some went to the "Matches" tab instead of the "Profile" tab
- People don't really view the Dashboard page other than when they first login
- "Very clean", "like the color scheme" "cool navigation bar"

<u>Specific comments/quotes on certain pages:</u>

Dashboard page

- Match calendar doesn't seem useful on the dashboard page
    - "Why not move the matches calendar into the matches tab?"
    - " I don't think I really need to see my profile card here"
- Motivational quote
    - "This is a great idea, especially aligns with your goals of promoting fitness. Would be nice though if you had a rotating quote."

Profile page

- "It's pretty clear and all the fields make sense"
- "It didn't take me too long"
    - On the contrary, another user said "The profile process was more intensive than I thought"
- "Thanks for including an additional info section. I have an injury and would like other to know that before they expect me to perform an exercise with them that I cannot do"
- "Unclear how descriptive the bio has to be"
    - "Is there any sort of filter on words (swearing, etc)?"
        - "Maybe you could add rights, rules and responsibilities?"
- "How would you know that my phone number is correct? Maybe include verification?"
- Cardiovascular fitness is too long, can be reduced to cardio
- "I like the current selection of interests, very focused on Stephen's fitness center, which is what I thought of when I heard GymBuddies"
- Maybe include more interests? (future work)
    - "I'd also like someone to just play basketball with at Dillon"
    - "What if I want to take group fitness classes with someone?"

Find a Buddy

- “Ohhhh, so I can see their profile and availability. I’m not just matching based on profile and then scheduling after? That’s nice.”
- Not so intuitive when it says “no more matches”
    - “Only had like 2 candidates and then ran out”
- “I couldn’t see the pass/match button because the calendar was too big”
- “I keep forgetting to click times to request them”
- “LOL match and pass… it’s like a tinder for the gym?”
- Generally a bit confused on what to do at first
    - some don’t read the instructions.
        - we had to tell them to read the instructions

Calendar UI comments

- “The color scheme and UI itself is very clean”
- “The drag feature is pretty convenient”
- “The ‘clear’ button that clears the UI for me is nice”
- Calendar UI view cons:
    - Can’t see full name of user you are matched with
    - No lines to indicate the time, so have to guess
        - “Is this time 7:30 or 8pm? I can’t really see.”
- “Ohhhh you have to click once and then hover? Interesting.”
    - “It took me like 2 tries but it wasn't hard to understand”
- “Woah i can like overlap two time blocks and it merges them”

Incoming Requests

- “Woah! It’s kinda fun to keep ping ponging requests back and forth to each other”
- “Modifying requests is probably the most helpful feature you have because it gives me more flexibility and really sets the match in stone since both people have a say in the final match times. Also lets you continue coordinating with someone you really like instead of having to delete the request because you don’t like the times”
- Incoming requests modal displaying both the modify requests button and accept button
    - “It would be nice if the ‘modify’ button only appeared after letting you modify.. If I just open up the request and see both buttons initially it’s confusing”

Settings

- “Love how you have SMS notifications… I don’t want to login to the app every time to check!”
- “The little in app notifications are nice and helpful so I don’t have to go between pages all the time to check if someone new has appeared”

Blocking

- “Oh it’s nice that you implemented this feature! I have some people I do not want to be receiving requests”
    - “It’d be awkward if I saw my ex on here”
- “Would be nice if I can block people that I encounter from the find a buddy page, but there’s no netid listed on the Find a Buddy page that I can use to search for and block people”

Deleting account

- “This is nice… I don’t really want my number and stuff floating around online”