Casey Klippel
Jiamin Xuan
Daniel Cathie
Runhui Miao
Deepali Bhanushali

Paymodoro Final Report

Executive Summary

Paymodoro is a productivity app that motivates users through financial buy-in. As people increasingly look for ways to break bad distraction habits, the popularity of productivity apps is soaring. We have developed a timer app that allows users to challenge themselves through making a financial payment through PayPal, and then setting a timer for focus periods. We ran into challenges throughout the development, and there are additional features we hope to implement, but our final app features all the basic functionality and a clean user interface.

Introduction

The human brain is not a multi-task machine and attention is limited. You need time to immerse yourself in your tasks and then structure your workflow. We have all these new technologies which are great at distracting us, but few that keep us on track. Our primary goal is to help users of any age and any background to focus on what they are doing right now- no matter whether they are preparing an exam, cooking meals, or even prepping for a date.

One study found that office distractions eat up an average of 2.1 hours a day. Another study, published in 2005, found that employees spent an average of 11 minutes on a project before being distracted. After an interruption, it takes them 25 minutes to return to the original task, if they do so at all. People switch activities *every three minutes*, whether making a call, speaking with someone in their cubicle, or working on a document.[1]

Description of the project

Paymodoro is a timer which implements the Pomodoro Technique, along with economic and social incentives, to help you keep away from your electronic devices and encourage you to focus on your task. Ideally, it consists of a SNS (Social Networking System), the Pomodoro Technique, along with powerful machine learning event classification.

In the future, you can open Paymodoro on your laptop, smartphone, or smartwatch and get started. First, you set up a timer with proper amount of time for your task and authorize a $2 payment to Paymodoro via Paypal. Next, Paymodoro will remind you wisely on when you should work and when you should take a break using the Pomodoro Technique. Then, we are going to use (after asking permission, of course) your cell phone accelerometer, Kinect motion detector, or touch pad to classify whether you are focusing on your task.

Market analysis

Customer Segment

It is a single-sided market opened primarily to university and high school students as well as company employees and self employed people. Essentially, anyone who needs to manage their time and to focus on their work is our potential customer. Students might need our products the most, particularly during exam times.

Our SNS and Paypal feature will give you a penalty for being distracted and thus makes your outcome (economic and social) of being distracted very low. The PayPal feature is currently implemented, while the SNS is left for version 2.0. The SNS will connect you to your friends thus you can check your friends' achievement and what they fail to achieve. Moreover, if you can check what other people around the world are achieving, their success might also motivate you achieve your goals.

Channels

In order to market this app, we would take to the channels used by our most visible target demo: college and high school students. In addition to traditional digital marketing tools like AdWords, and Facebook advertising, we would push content to platforms such as Snapchat, Reddit, Yik Yak, and Twitter. We would use sponsored "snap stories" in snapchat for raising general awareness of Paymodoro, leveraging our mascot aggressively in advertisement. While Yik Yak has only just begun its business partner program, we believe the market segments that we target would align with its exploding user base, also increasing our reach.

We aim to utilize Twitter and Reddit as a means of building an active user base around Pomodoro, encouraging our followers to share their ideas for new functionality. We would aim to poach users from subreddits like r/apps, r/tech, among others, as well as Twitter users who follow similar app companies. Once/if we have established a dedicated community with integrated social features, we could release our code, further prompting new modifications and suggestions.

Some further ideas:

- Utilize our subreddit for customer service info, including Patch/Update notes, Deals/Discounts/Announcements.
- Potentially crowdfunding, angel investors, or seed-funding talks to switch hosting services to other companies and move off the NYU server (next steps)
- Starting offers of free wagers to attract users (ala the "Uber Approach")
- Independently-managed wagering business units; for example, one that profits off the community pool, and one that profits off the wagering with friends pool, to avoid regulation
- Set up corporate accounts to allow employees to be more productive on company time
- Collaborating with productivity experts and gurus, or productivity magazines (eg Real Simple)

Customer relationship

We keep customers by using economic incentives and social networking.[1] 88% percent of people fail to achieve their goals and thus we will just reward the rest 12% people. Once you've achieved your goal, we will send you initial buy-in, as well as a small amount of money as your "winnings." Because people are interested in improving their habits and keeping up with their friends, we can leverage human nature to change distraction habits and hope to keep user engaged.

If customers complain about mistakenly being charged, we will have an immediate response to those complaints. Overall, user experience should be positive, because Paymodoro is investing in yourself.

Key partner

Customer partners: parents of young students, employers, high schoolers, universities,co-working spaces

Business partners: PayPal, Facebook (people may think Facebook will distract them, but checking what other friends are achieving is fun and inspiring).

Revenue stream

We projected financials for Paymodoro. Revenues are entirely derived from customer's challenge buy-ins. Costs include: Payouts of principal and premiums to successful players, AWS hosting fees, API licenses, and developer salaries.

Projected Income Statement for Paymodoro. (Assumption Table Below)

Revenues	$ 10,656,145	$ 53,813,533	$ 108,703,336	$ 548,951,847	$ 831,662,048	$ 1,119,971,557
COGS: Success Payout, Principal	$ 5,328,073	$ 26,906,766	$ 54,351,668	$ 274,475,923	$ 415,831,024	$ 559,985,779
COGS: Success Payout, Premium	$ 799,211	$ 4,036,015	$ 8,152,750	$ 41,171,388	$ 62,374,654	$ 83,997,867
COGS: Hosting Costs	$ 210,992	$ 1,065,508	$ 2,152,326	$ 10,869,247	$ 16,466,909	$ 22,175,437
Total COGS	$ 6,338,275	$ 32,008,289	$ 64,656,744	$ 326,516,558	$ 494,672,586	$ 666,159,082
Gross Profit	$ 4,317,870	$ 21,805,243	$ 44,046,592	$ 222,435,288	$ 336,989,462	$ 453,812,475
Gross Margin	*41%*	*41%*	*41%*	*41%*	*41%*	*41%*
SG&A	$ 3,000,000	$ 5,000,000	$ 10,000,000	$ 15,000,000	$ 20,000,000	$ 30,000,000
EBIT	$ 1,317,870	$ 16,805,243	$ 34,046,592	$ 207,435,288	$ 316,989,462	$ 423,812,475
EBIT Margin	*12%*	*31%*	*31%*	*38%*	*38%*	*38%*
Taxes	527,148	6,722,097	13,618,637	82,974,115	126,795,785	169,524,990
Net Income	$ 790,722	$ 10,083,146	$ 20,427,955	$ 124,461,173	$ 190,193,677	$ 254,287,485
Net Margin	*7%*	*19%*	*19%*	*23%*	*23%*	*23%*

[1] https://blog.bufferapp.com/the-science-of-new-years-resolutions-why-88-fail-and-how-to-make-them-work

Notice that the margins are very high. However, they are not unreasonable for the business model. The gaming industry has pre-tax operating margins of 17%[2], and Paymodoro does not have any of the depreciation & amortization associated with physical casinos. Additionally, Paymodoro should have lower SG&A costs than physical casinos.

Assumptions Driving Paymodoro Income Statement

	2015	2016	2017	2018	2019	2020
Number of US univeristy students	21,000,000					
Rest of world student adoption	21,000,000					
Number of freelancer workers, US	53,000,000					
Rest of world freelancers	53,000,000					
Growth of target market	1%	1%	1%	1%	1%	1%
Average % success	50%	50%	50%	50%	50%	50%
% of Loser Pool Paid to Winners	15%	15%	15%	15%	15%	15%
AWS cost per visit per month	$ 0.0033	$ 0.0033	$ 0.0033	$ 0.0033	$ 0.0033	$ 0.0033
AWS cost per visit per year	$ 0.04	$ 0.04	$ 0.04	$ 0.04	$ 0.04	$ 0.04
Number of Visits per user per month	30	30	30	30	30	30
Number of User Visits per Year	360	360	360	360	360	360
SG&A Costs	$ 3,000,000	$ 5,000,000	$ 10,000,000	$ 15,000,000	$ 20,000,000	$ 30,000,000
Tax Rate	40%	40%	40%	40%	40%	40%
Total Target Market worldwide	148,002,015	149,482,035	150,976,856	152,486,624	154,011,490	155,551,605
Percent of Market Paymodoro captures	0.01%	0.05%	0.10%	0.50%	0.75%	1.00%
Total Paymodoro Users	14,800	74,741	150,977	762,433	1,155,086	1,555,516
Average Challenge Payment Size	$ 2	$ 2	$ 2	$ 2	$ 2	$ 2

Additional Sources and Assumptions

Sources and Assumptions
http://nces.ed.gov/fastfacts/display.asp?id=372
https://www.upwork.com/press/2014/09/03/53-million-americans-now-freelance-new-study-finds-2/
http://blog.heapanalytics.com/how-we-estimated-our-aws-costs-before-shipping-any-code-2/
D&A=0, and Interest Expenses=0.

[2] http://pages.stern.nyu.edu/~adamodar/New_Home_Page/datafile/margin.html

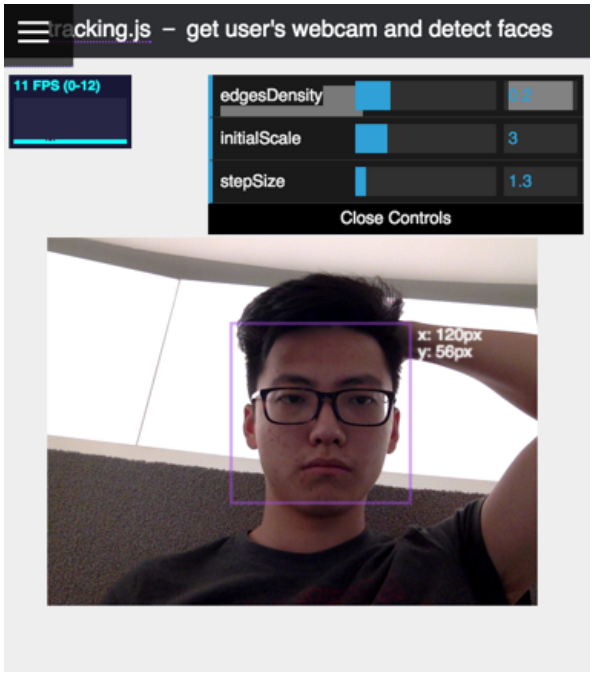
Problems encountered

1. Distraction detection

The most important feature and the most frequently asked question is about the distraction detection. There are many great ideas we can use:

The easiest one is detecting the mouse movement or switching an app, even detecting the acceleration of your phone when you try to pick up your phone. The only method that we have implemented as of now is warning users when they leave the app, but it obviously has many problems in telling whether you are actually distracted, or if you've accidentally exited or have a legitimate task (ie, an important incoming phone call). Although in the future we can set up a proper logic to tell whether users distracted and set up a fast refund workflow, it will be a nontrivial process.

Another method we anticipate adding is measuring distraction by using facial recognition. It is especially a case for a task such as reading a book, so when you raise your head to look at a screen, your face will be detected, which will be regarded as distracted.

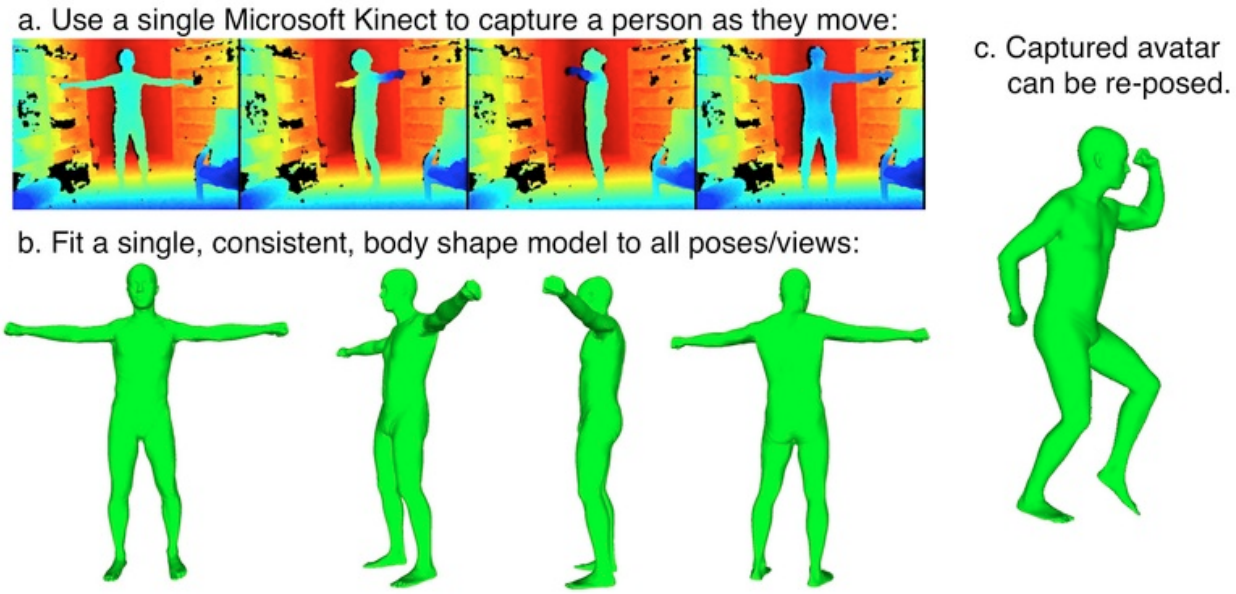


(As you can see in the picture, my face is detected even the light condition is not so good. It also works even when I make faces, but I won't add those photos here.)

But this method is also only good for some cases. It can never tell whether you are watching a episode of your favorite TV show, or legitimately looking at a Coursera class.

Another option is using the Apple Watch to distinguish your different patterns in different kinds of task using your pulse, hand movements, and GPS. And if you have a smart TV or kinect in

your room, we can also use that device to tell your body movement and use a recurrent neural network to classify your status over a period of time.



Basically it is like you hired a personal coach. It may sound like stories from *Black Mirror,* and people might say it is creepy, but it is not that far removed from daily life and we should be prepared for such changes in the near future.

2. Database connection issues

Building the database and server-side scripts, and separating the client-side scripts for use in Phonegap, was difficult. Also, in editing users' profiles, sometimes the changing password action will not properly update in the database. It is may be because the compatibility between phonegap and the database is not as perfect as we think, or other server script issues.

3. Version control

We ran into version updating delays that sometimes confused our team members. Some of the members worked on something which turns out to have already been done by another member. A little time was wasted.

Synchronization is very important. We could improve in keeping everybody informed about the status of the progress; this is a challenge most development teams likely encounter. Also, telling others about the obstacles we have run into in some fields is also important. That way, the team can decide whether to put more time into the issue or just put that feature on the back burner.

4. Security of user information

The information of the user will be saved in cookies in order to keep "logged in" status. That means it should be encrypted by the system, to prevent the user information from being easily stolen. We need to protect our users' information by encrypting our cookies' data.

Moreover, in order to add more security to the data, we will need to add an email link that will send a confirmation email when people register and change their profile.

5. Legal issues

Another big issue we are facing is legality. Because of several different state laws on the books, and depending on what state we are headquartered in, Paymodoro may or may not be subject to gambling regulations and restrictions. This could prove especially true if we pursue the Social Wagering functionality as a next step. As friends could bet against one another, Paymodoro could be potentially used illegally and become a mobile "bookie."
Additionally, due to potential bugs, users could lose their money if the app crashes and kicks them to the home screen, making us liable for not delivering on our product.

Fortunately, most states' gambling laws have provisions for "social gambling," excusing our company from liability, so long as we do not skim a profit from this peer-to-peer wagering. Additionally, most bug- and payment-related issues could be avoided by allowing customers to retrieve their lost funds through a customer service mechanism. This would preferably be initiated through contact on our social media platforms. While we will honor all cases made in good faith, we will monitor for suspicious or repeated behavior, and will be able to lock accounts looking to game the system. The goal would be to make the process transparent and fair, yet also thorough, in order to prevent revenues from disappearing.

Lessons learned

1. jQuery system action

Using the jQuery framework helps us to monitor system events easily. Our task is to monitor the mouse or touchpad movement. So after you click the start button, you can't move your mouse anymore otherwise you will be charged. However, jQuery only functioned in the web browser, so if you closed the browser or open a new tab, jQuery can do nothing about that.

```
//change color and stop the timer
$('#pomodoro25').mouseleave(function() {
    timerStop();
    var p =$('.wrapper').css("background", "#b25244");
});
```

But we can use javascript EventListener touchstart and touchend to detect tough movement on phone.

```
<script type="text/javascript">
  function onload() {
    document.getElementById('touchstart').addEventListener('touchstart', hello, false);
    document.getElementById('touchend').addEventListener('touchend', bye, false);
  }
```

2. PHP working with AJAX&JSON

PHP can not be embedded in phonegap, so we use Json to transport the data between the server and clients. If we just receive the data from an “echo” by PHP, we can not get any additional message echoed from the PHP file, which will cause an error if it is not encoded in the json format.
In PHP, we found that arrays were much easier to use. We can create messages or flags and simultaneously put them into the growing array. And we can use a two-dimensional array to save many fields of user data and encode it in the json format. Then, we use the user’s information by decoding the json data sent by our PHP file, which interacts with the database.

3. Phonegap plugin.
One challenge is PhoneGap’s lack of file system recognition. We tried to organize the code in separate html/, css/, js/, and images/ directories. This clean structure works well on the web, but PhoneGap needs all the CSS, JS, and images to be in the same folder as the html pages. Debugging that issue caused some headaches. For example, if the index.html file is stored in the html directory (ie, html/index.html), phonegap cannot find an image (eg, tomato.png) stored in the images folder (ie, images/tomato.png), because it can’t interpret the location “../images/tomato.png”

Phonegap Build is a great online framework because it leverages the benefit of cordova and allows you to use html, css, javascript to create app on different platforms simultaneously. It also has many plugins to detect phone movement. But since it is an online platform, debugging is not that easy. We’ve been using device-motion and dialogs plugin to detect whether a user is putting the app into background, and to determine whether we will send notification. There is a trivial problem with that task that there are two kinds of userCapture phase, capture phase or bubbling phase. Bubbling is used after an event triggers on the deepest possible element, it then triggers on parents in nesting order. Capture phase is just the opposite. In our case we should use capture phase instead of bubbling phase.

4. PayPal API
The PayPal Sandbox for creating test businesses and payers is sometimes buggy. It is very closely modeled on the real PayPal website, which is usually an advantage. However, sometimes it is too closely connected - that is, links from Sandbox go directly to the real PayPal site, which causes test development to break. Also, the UI in the Sandbox site has change since PayPal wrote its documentation, leading to frustrating searches (or visits to Stack Overflow).

Next Steps

There are several directions that could be fruitful expansion strategies for Paymodoro:

- **Apple Watch/Kinect**. One of the most popular apps on the Apple Watch now (in its debut weeks) is a streamlined cooking timer app (see “Kitchen Stories” and “Boiled Egg Timer”).[3] Paymodoro is a productivity timer, and thus it’s in the same vein as

[3] http://www.wareable.com/apple-watch/best-apple-watch-apps-832

these popular apps. We can use the “taptic” engine to remind people to focus. Also, its behavior tracking abilities can help people learn about their habits (ie, when are they distracted, versus when are they productive).

- **Social Wagering.** The idea is to leverage Paymodoro so you can bet against your friends. After sending them an invite, you can challenge your friends to complete a specific task, or you can see what tasks your friends fail to achieve. An social-recognition award system can also be added to reward you for your achievements, or for your accumulated time focusing (measured by how much time you have successfully logged in the timer portion of the app). We may need to set additional confirmation messages to occur at every “break” period to prevent people from setting Paymodoro to run while they sleep.
- **GPS location**. The idea is to see how people are faring in their productivity challenges near you. Example: can you beat the average NYU student’s productivity in Bobst during finals week?

Conclusion

The market is ready for an app such as Paymodoro. Our current version delivers basic functionality. With additional features, the app can become even more sophisticated at tracking user behavior and responding to user needs. The Paymodoro model is also quite scalable, paving the way for future growth.

Phonegap is a revolution in the mobile development, making it easy for people to develop mobile apps. By using it, we have successfully developed our app, Paymodoro. However, it still has some limits, so this technique does not have exactly the same powerful capabilities as developing through an Android or iOS SDK. But to launch our v1.0 app, by using phonegap and honing our development skills, we will make Paymodoro more and more popular.

Appendices
Appendix Part 1: Code Locations

The code for the entire app is contained in the paymodoro_master directory at this URL:
http://websys3.stern.nyu.edu/websysS15GB/websysS15GB2/public_html/websys/

Alternatively, navigate directly to paymodoro_master at this URL:
http://websys3.stern.nyu.edu/websysS15GB/websysS15GB2/public_html/websys/paymodoro_master/
The two subdirectories in the paymodoro_master directory contain all the parts of the app. The php directory contains all the php code, and the paymodoro directory contains all the HTML, CSS, and JS. The app is built from the paymodoro directory (actually, from a GitHub repo that contains exactly the same contents) using PhoneGap Build.

To launch Paymodoro as a web app, visit paymodoro_master/paymodoro/html/ at this URL:
http://websys3.stern.nyu.edu/websysS15GB/websysS15GB2/public_html/websys/paymodoro_master/paymodoro/html/

To download the Paymodoro APK, download navigate to this URL and click on paymodoro.apk:
http://websys3.stern.nyu.edu/websysS15GB/websysS15GB2/public_html/websys/
Alternatively, navigate directly to the below URL and the APK will download automatically:
http://websys3.stern.nyu.edu/websysS15GB/websysS15GB2/public_html/websys/paymodoro.apk

Appendix Part 2: App Functionality and Screenshot Walkthrough

This section features an overview in list form of the Paymodoro's functions. Then, it provides a walkthrough with screenshots.

The code has built-in redirect times of 3 to 5 seconds for several of the pages. While a user is being redirected with a wait time, a message will display that the redirect will take a few seconds.

App Functionality: What Users Can Do in Paymodoro

1. On the login screen, users can register for an account, or login if they already have an account. Let's assume the user needs to register for an account. They will click on the "Register Here!" link.
2. On the registration page, the user enters their choice of username, password, and email. If they choose a username that already exists, an error message will prompt them to choose a different one. Once the user hits "Submit," their registration will be saved and they will be redirected to the login screen.
3. Back at the login screen, the user can now login with their username/ password combination.
4. Now the user is at the home screen. There are four options on this screen:
    a. Buy a Challenge Amount under the heading "Let's Get Set!"
    b. Edit my profile under the heading "User Tools"
    c. Logout under "User Tools"
    d. Tweet #Timeowner at the very bottom.
5. If the user chooses (b) "Edit my Profile," they will be brought to the profile editing screen. There, they enter their existing username and a new password and email address. Once they submit, they are redirected to the home screen.
6. If the user chooses (c) "Logout," they will be logged out and redirected to the login page.
7. If the user chooses (d) "Tweet #Timeowner", they can post the following message to Twitter: "I have been focusing with Paymodoro!"
8. The primary choice is (a) "Buy a Challenge Amount." Users can choose their buy-in level for using Paymodoro. There are three choices: Small Tomatoes ($1), Medium Tomatoes ($2), and Large Tomatoes ($4). Once they select the amount, they tap "Buy Now" to checkout with PayPal.
9. The user is redirected to PayPal to checkout. For the beta release of Paymodoro, we have implemented a "test store" and a "test purchaser" account. That way the user can test the app functionality without actually paying real money. The login credentials for the "test purchaser" are username: payer1@paymodoro.com and password: buytomato. These credentials are also displayed on screen in case you forget.
10. The user checks out through PayPal, and is automatically redirected to the timer screen in Paymodoro.

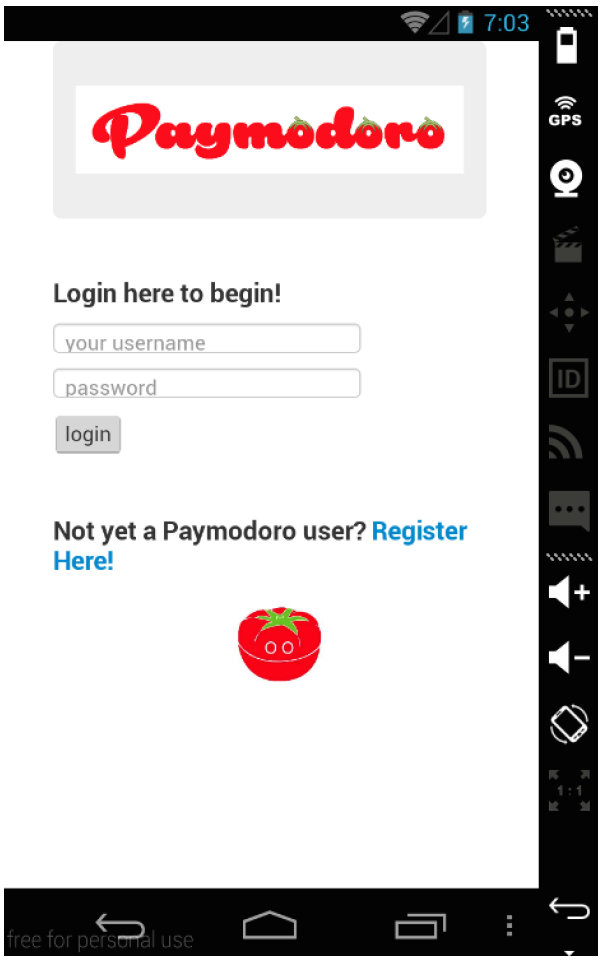
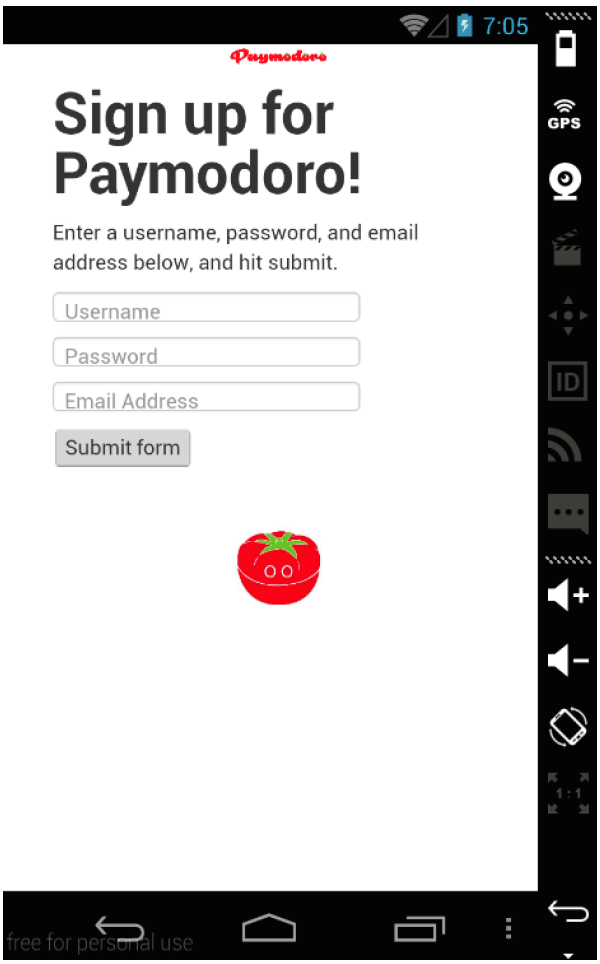
11. On the timer screen, the user can pick how long they want to focus. They can choose a length of time from the dropdown menu. In this beta release, they can also choose a demo time from the “Minutes” dropdown.
12. The user clicks “Start” once they’ve chosen their desired time. Each half-hour cycle starts with 25 minutes of “focus” time and ends with 5 minutes of “break” time. Messages are displayed on screen to note the focus and break times.
13. If the user tries to navigate away from Paymodoro via the back button while the countdown clock is running, the app will display a warning message.
14. If the user makes it until the end of the countdown timer, a congratulatory message will be displayed. At this point, the user can logout. Once they log out, they will be redirected to the login screen.

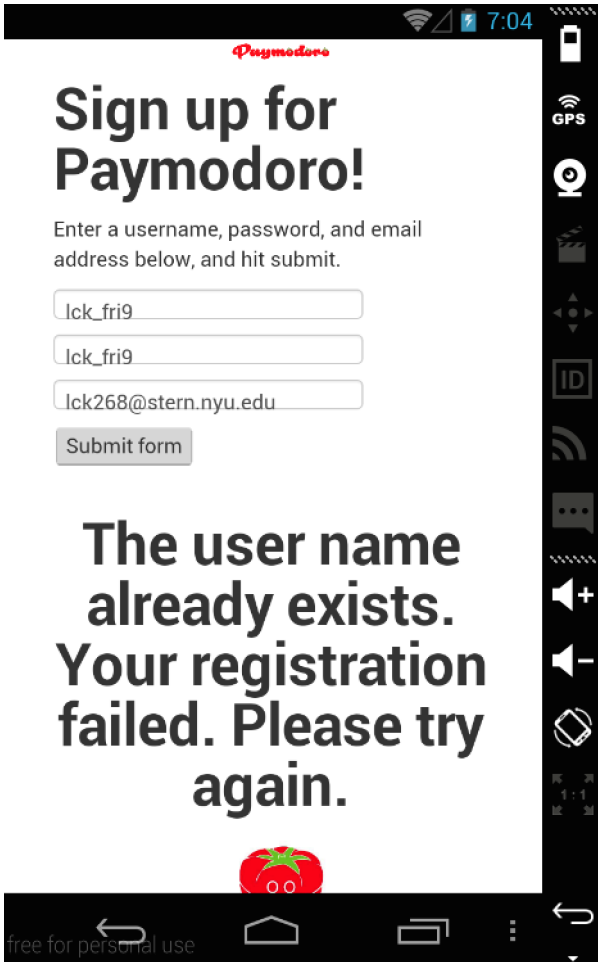
Screenshot Walkthrough

This section displays a tour of the app with screenshots. If you would like to follow along and use the app, there are a couple tips that will help your experience:

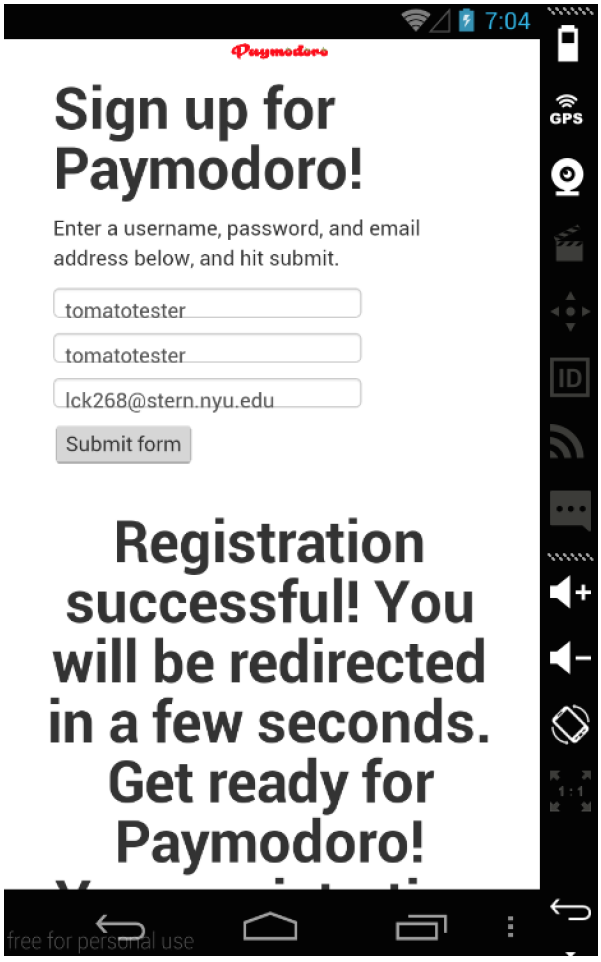
1. There are often 3-5 second redirect times built into pages in the app. A message will be displayed asking you to please wait while you are redirected. Please have patience!
2. Occasionally the PayPal Sandbox server gets overloaded and will not process the payment. Usually its issues will resolve within less than a minute (often in just a few seconds). If PayPal doesn’t process the request, please try again after a short break.

Let’s walk through the app. Feel free to register for an account or use a demo account (username: tomatotester password: tomatotester).

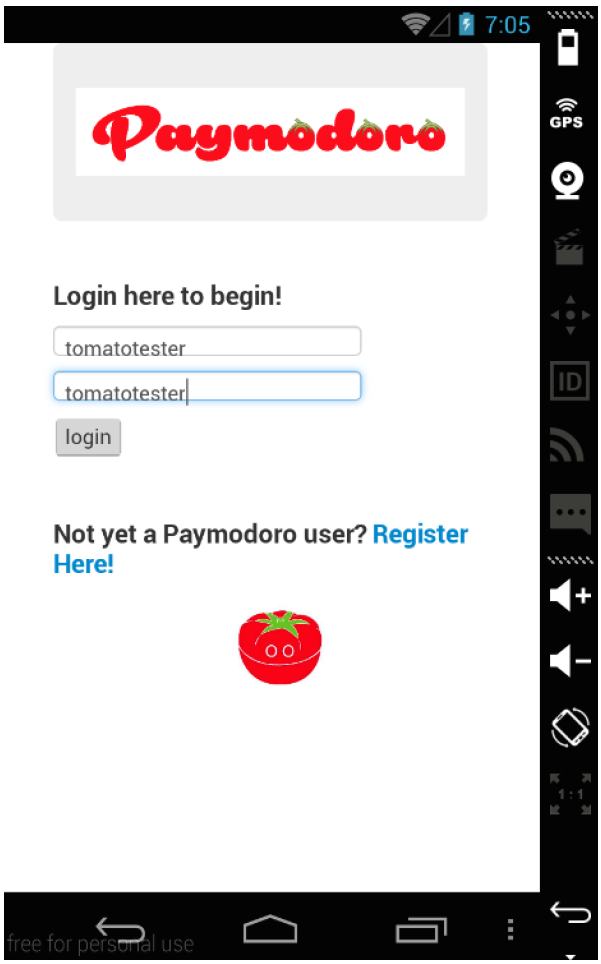
	
1. The login screen	2. User decides to register. Blank registration screen.



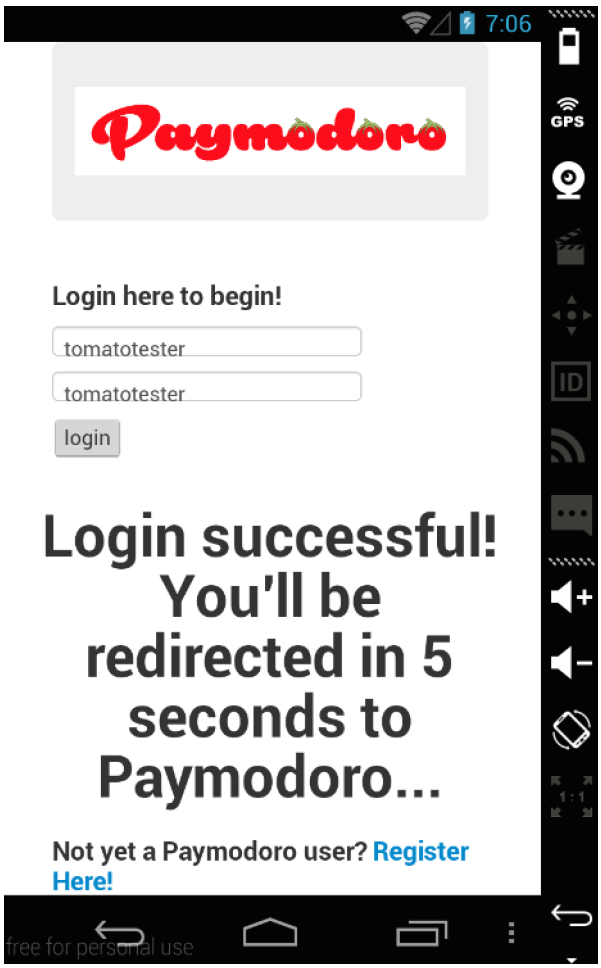
3. Failed registration example. Username is already taken.



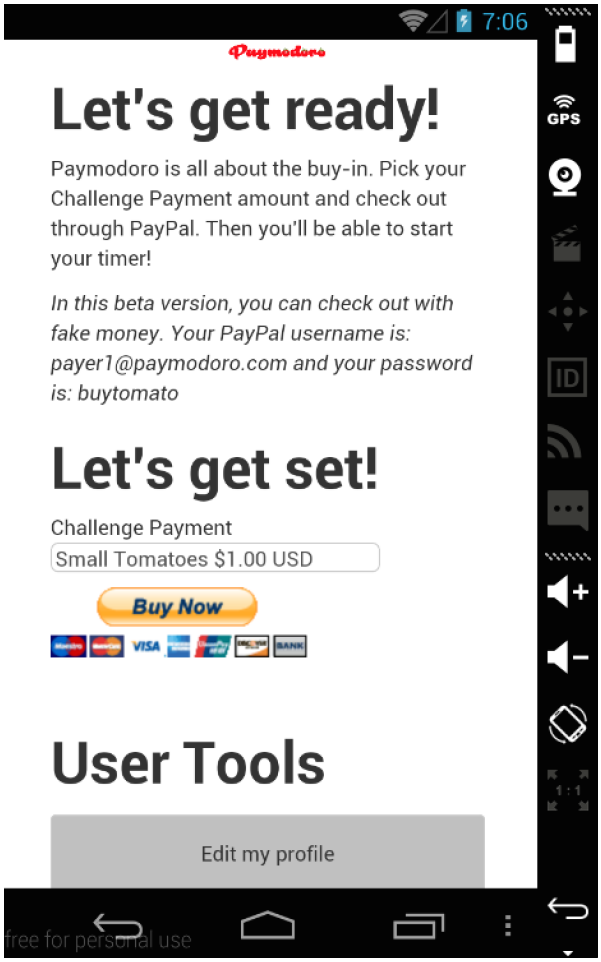
4. Successful registration example. Redirected back to login page.



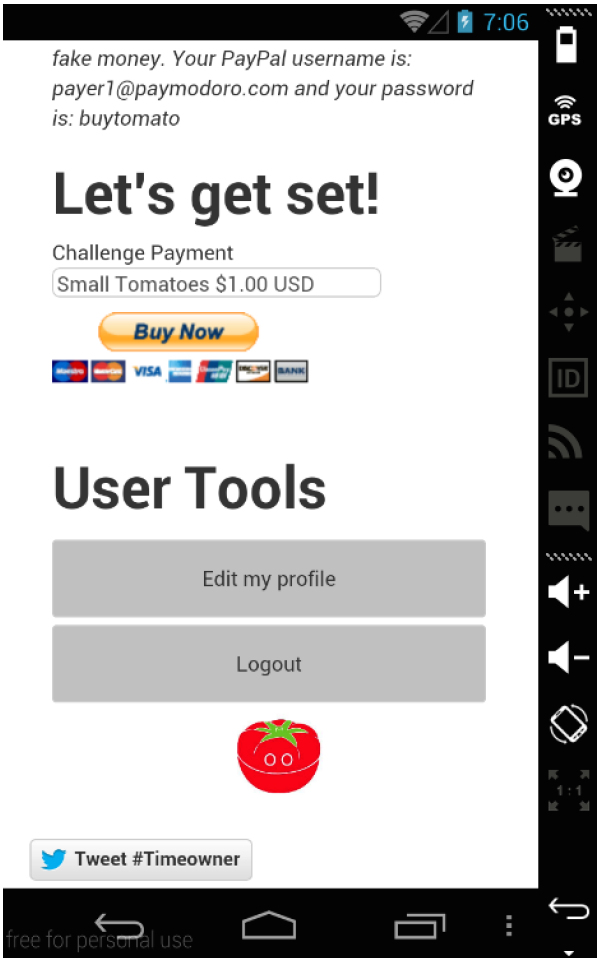
5. About to login



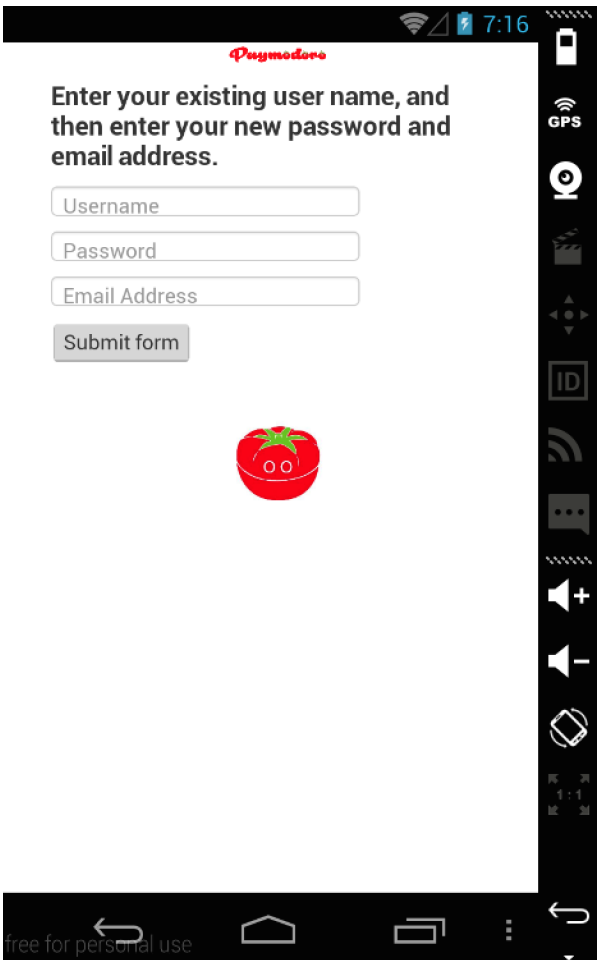
6. Successful login. Waiting for redirect.

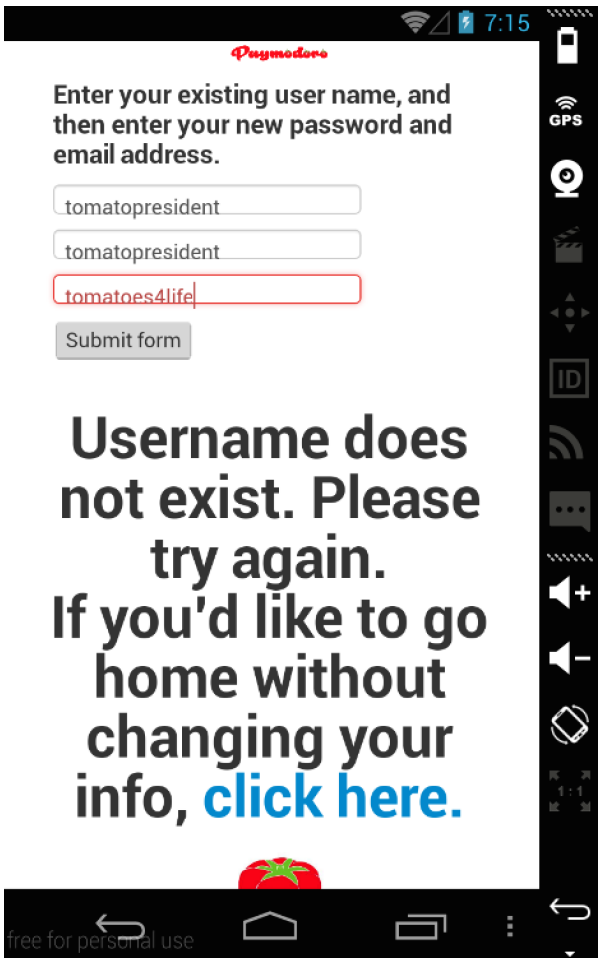


7. View of the homescreen (top portion).

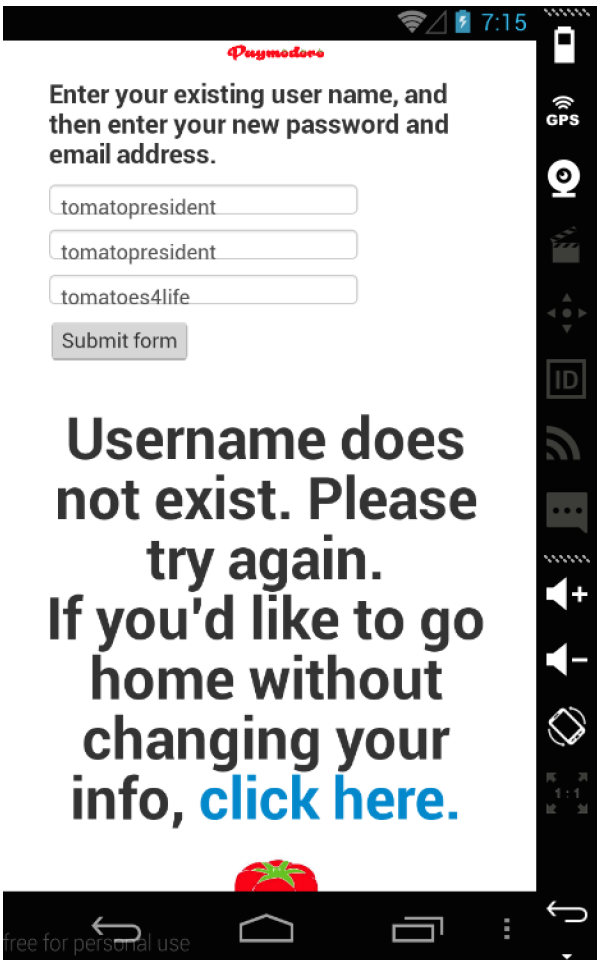


8. View of the homescreen (bottom portion).

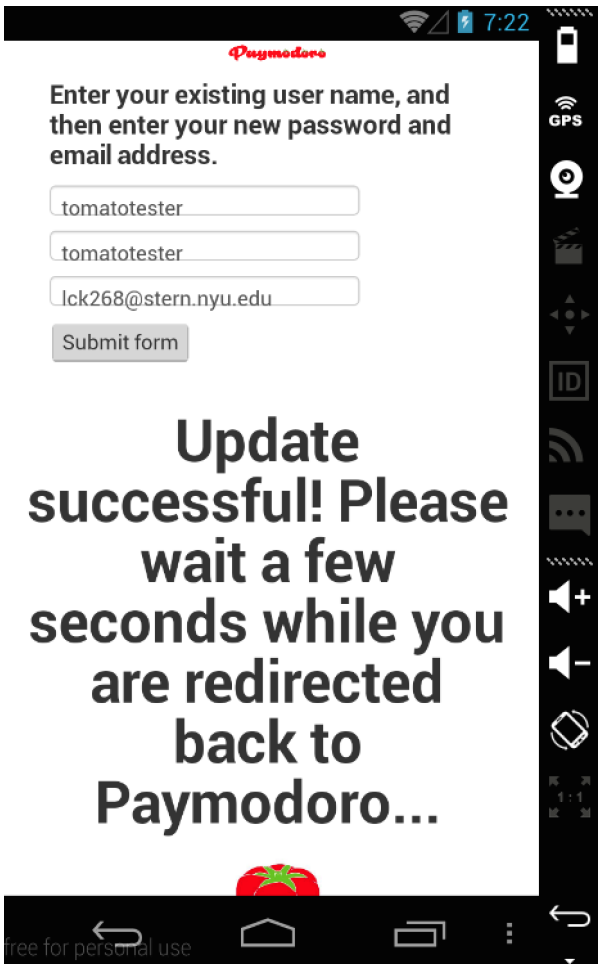
	
9. Redirect to “Edit my Profile” from the homescreen button “Edit my Profile” under “User Tools”	10. View of the edit my profile page.



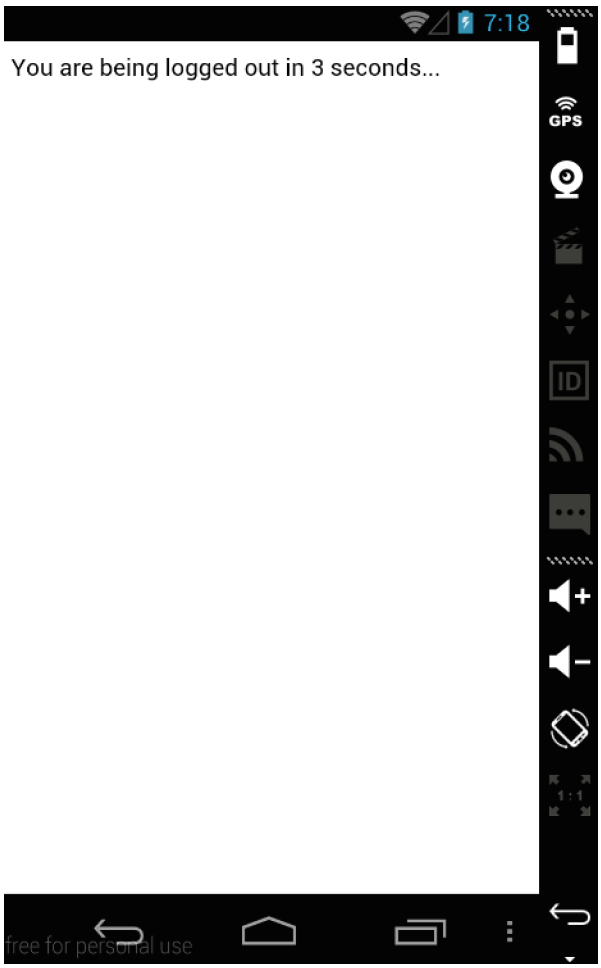
11. Example of highlighting for an incorrectly formatted email address. Emails must be in the format foo@bar.com



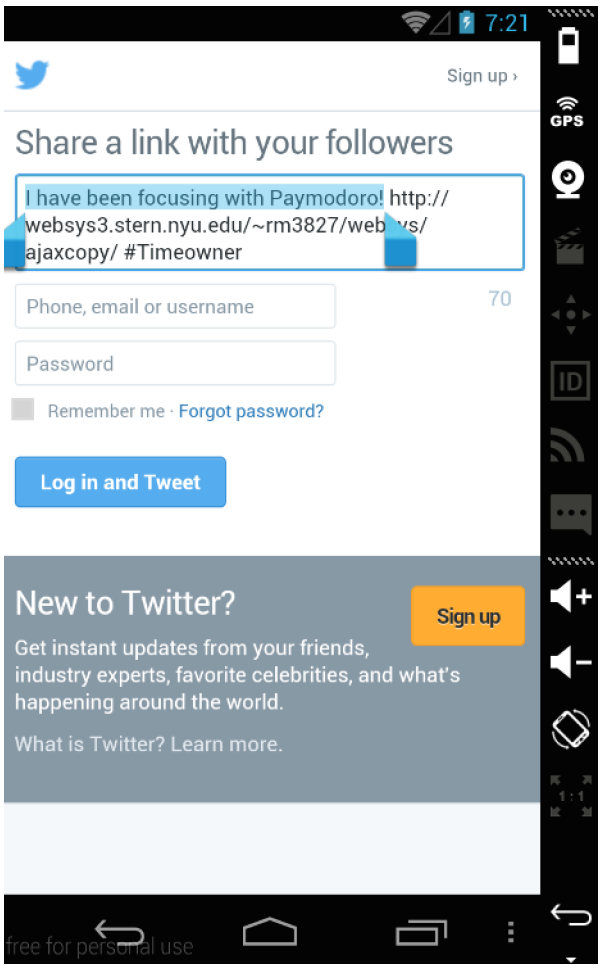
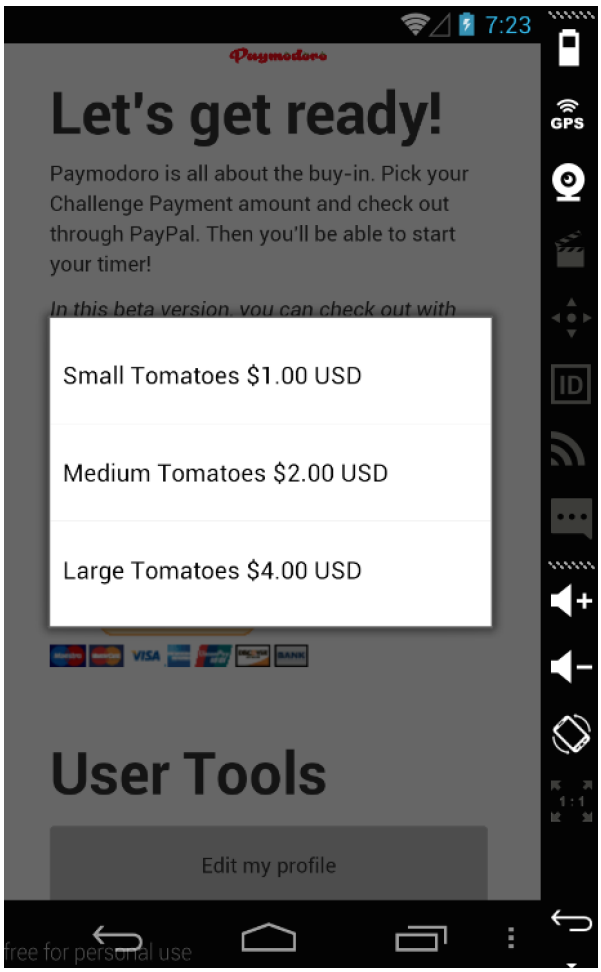
12. Example of a failed update, because the username does not exist. Updates have to be made for existing usernames.

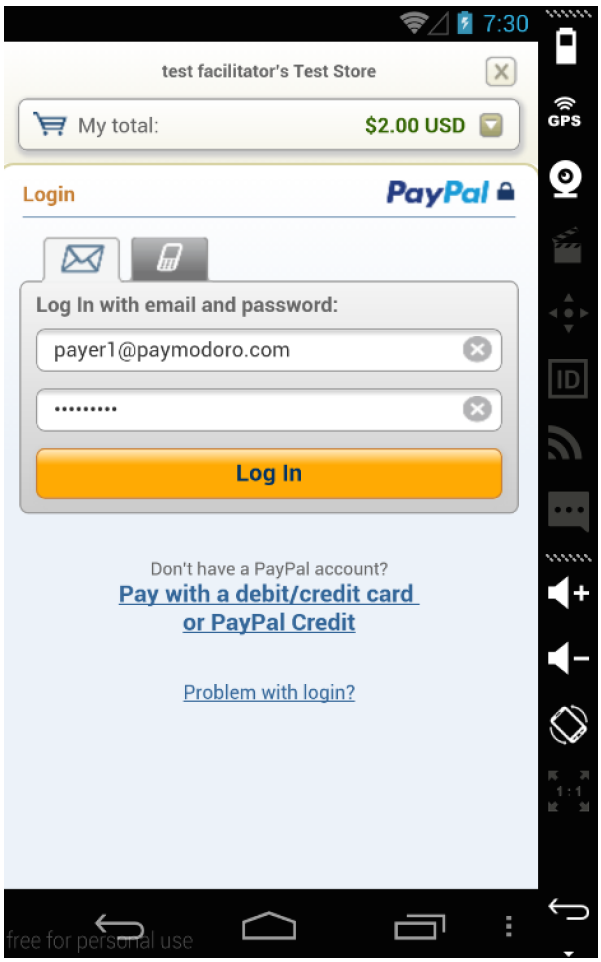
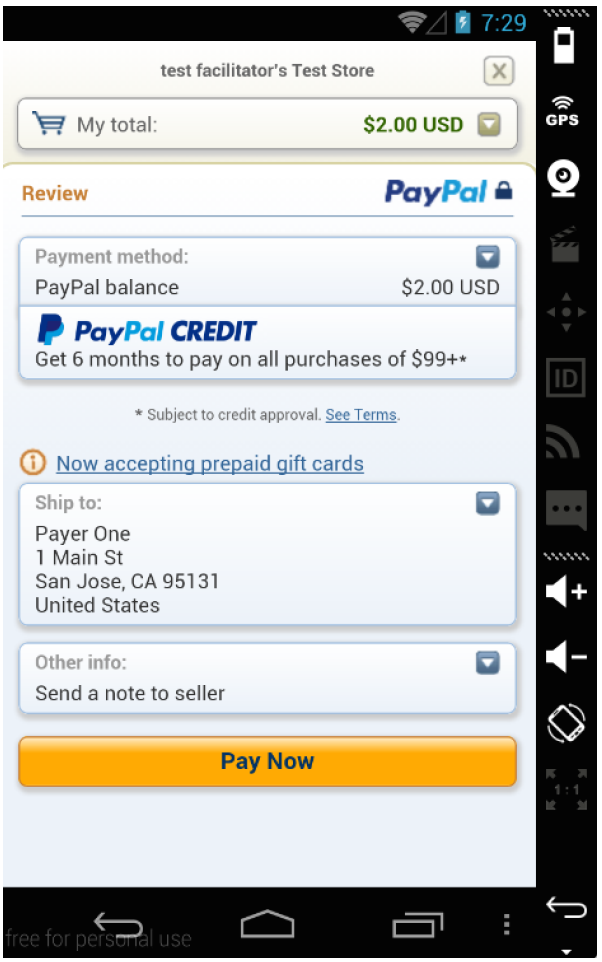


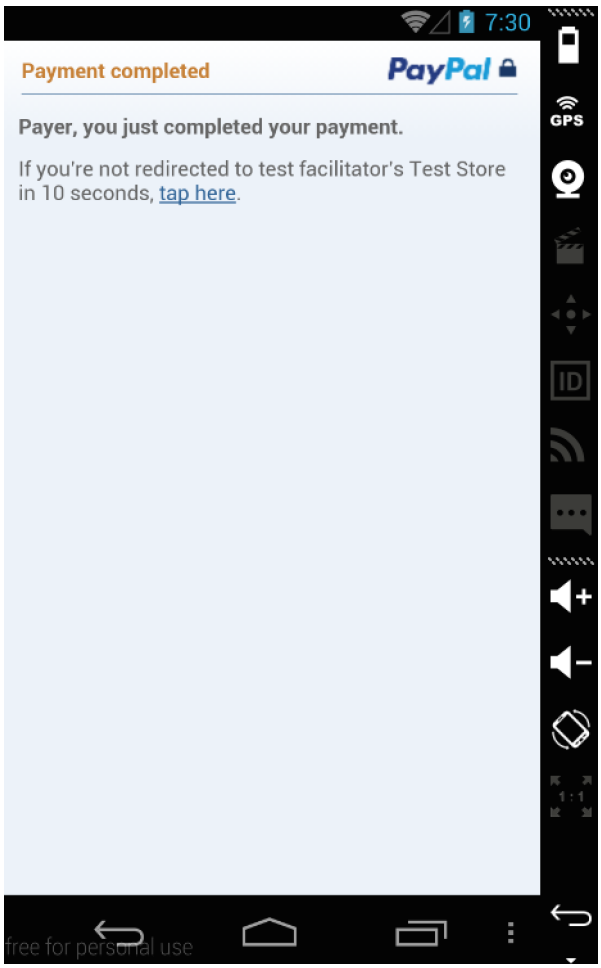
13. Successful update of profile. User is being redirected back to home screen.



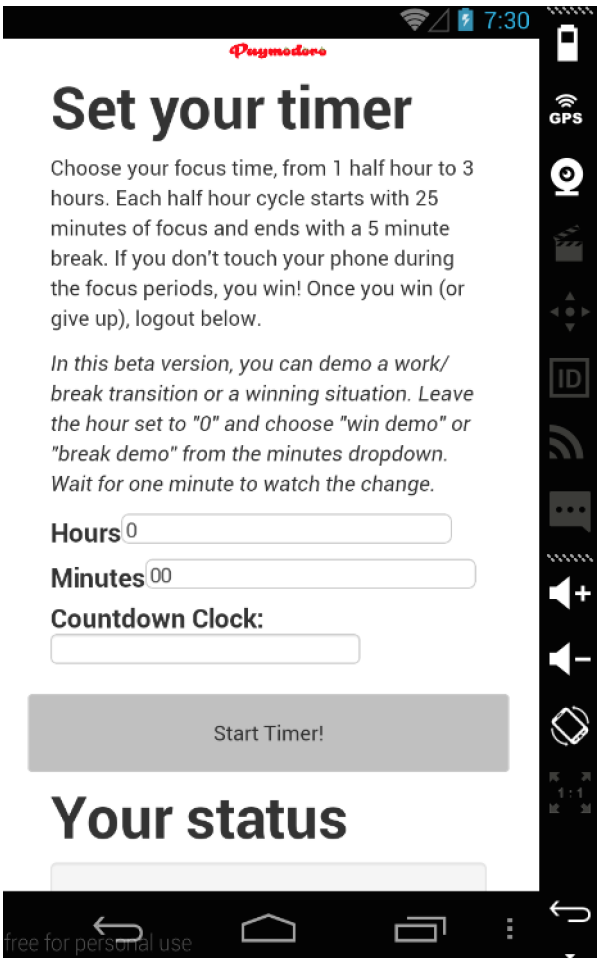
14. From the home page (#7 and #8), if the user choose to logout, they will see this logout screen.

	
15. From the home screen (#7 and #8), if the user chooses to tweet, they will see this page.	16. On the home screen (#7 and #8), the user can choose the challenge amount they want to pay by tapping on the drop down menu. Once they've selected an amount, they should tap the "Buy Now" button.

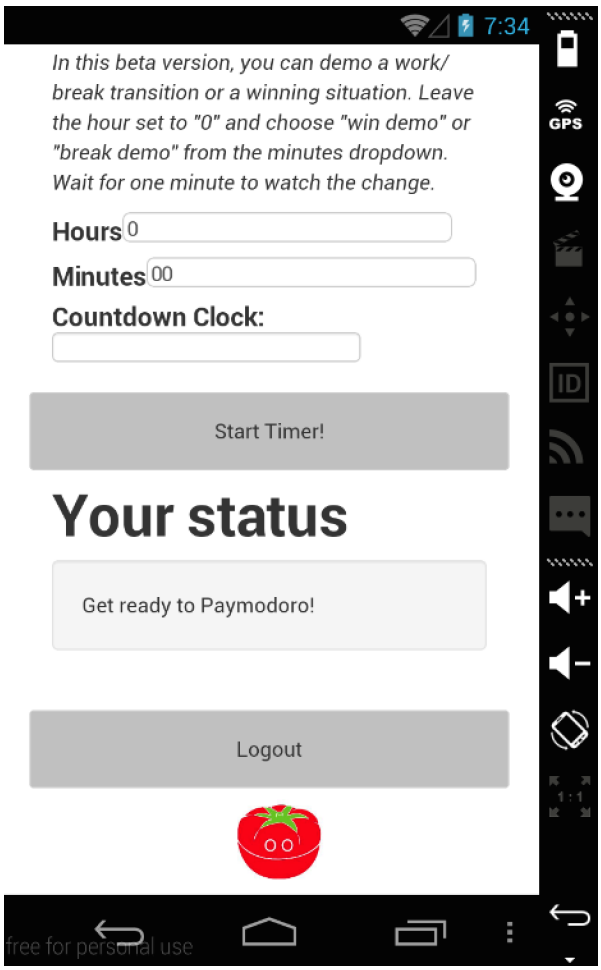
	
17. The user is redirected to a PayPal checkout site. In the beta version, the “test buyer” credentials should be used.	18. Once the user logs into their PayPal account, they tap “Pay Now” to checkout.



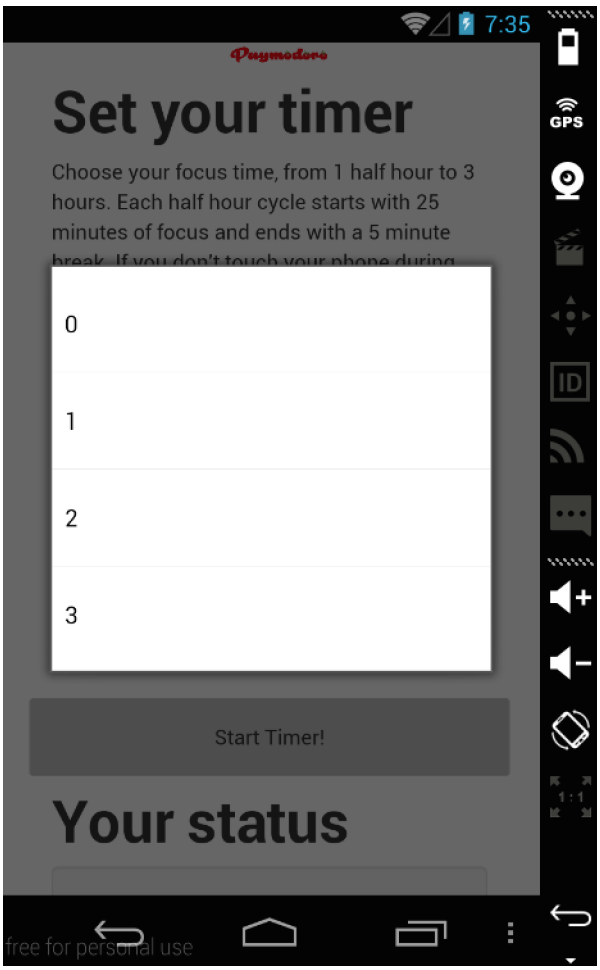
19. Once the payment is complete, the user will be automatically redirected to the timer screen.



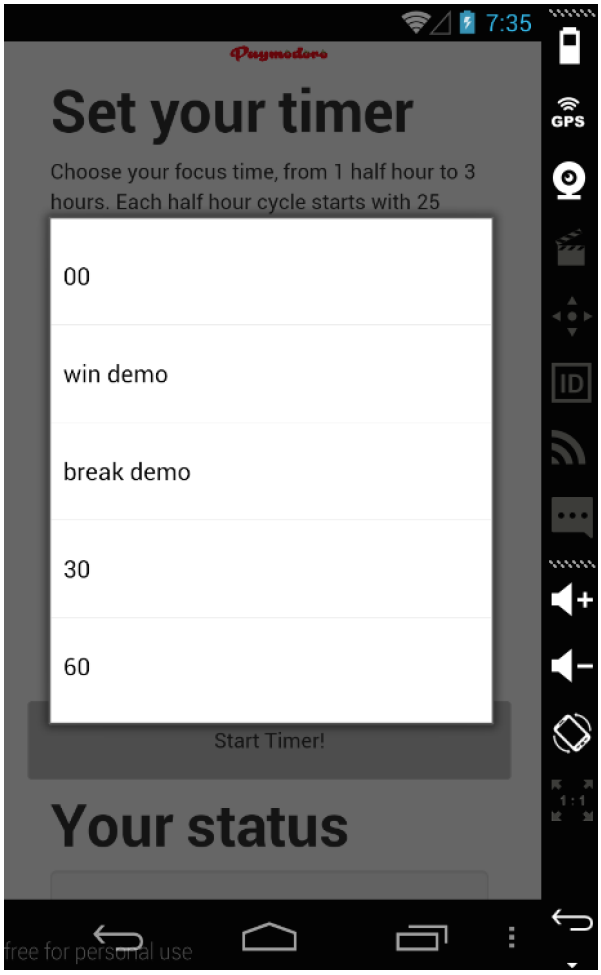
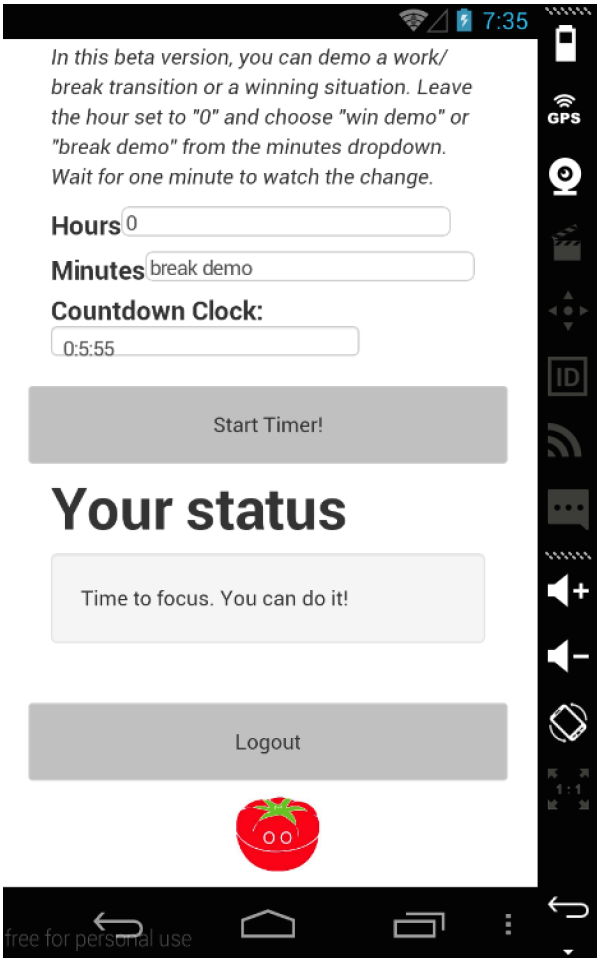
20. The first view of the timer screen (top portion).

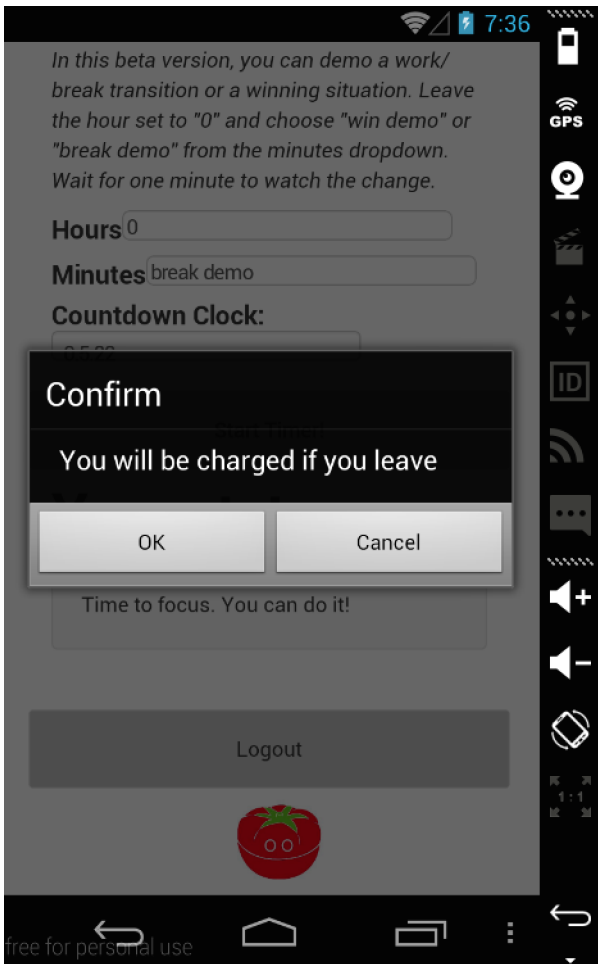
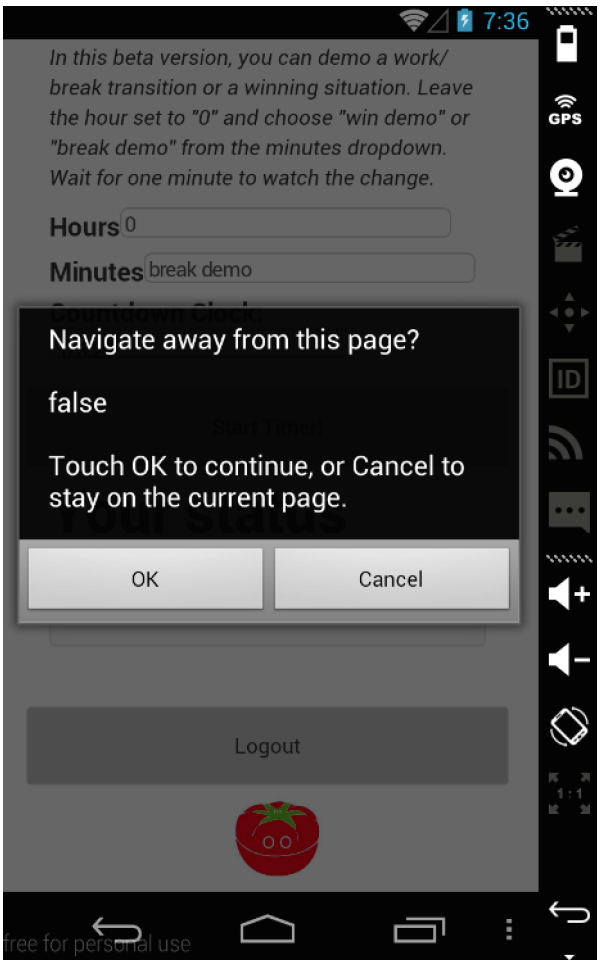


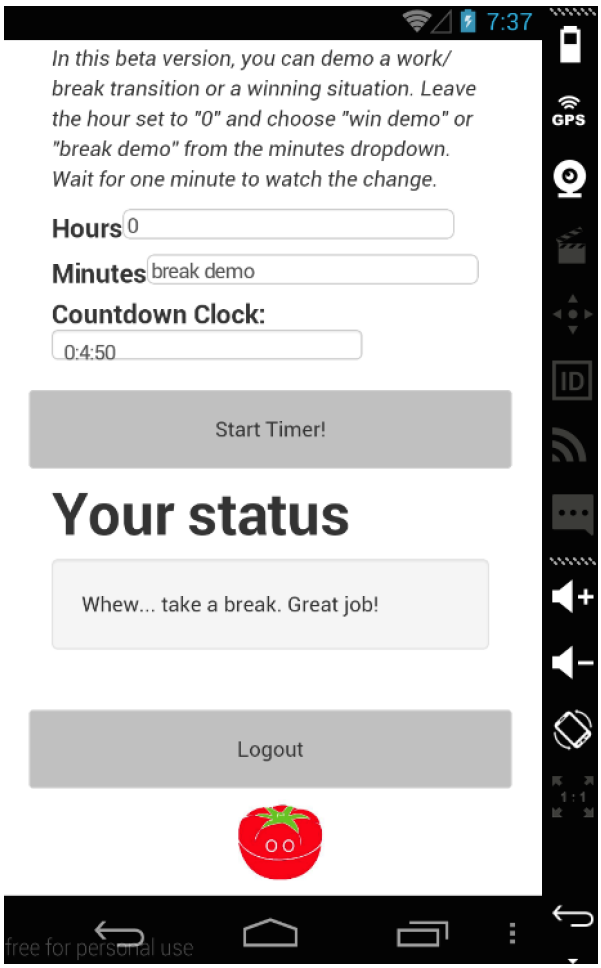
21. First view of the timer screen (bottom portion). Notice the status is “Get ready to Paymodoro!”



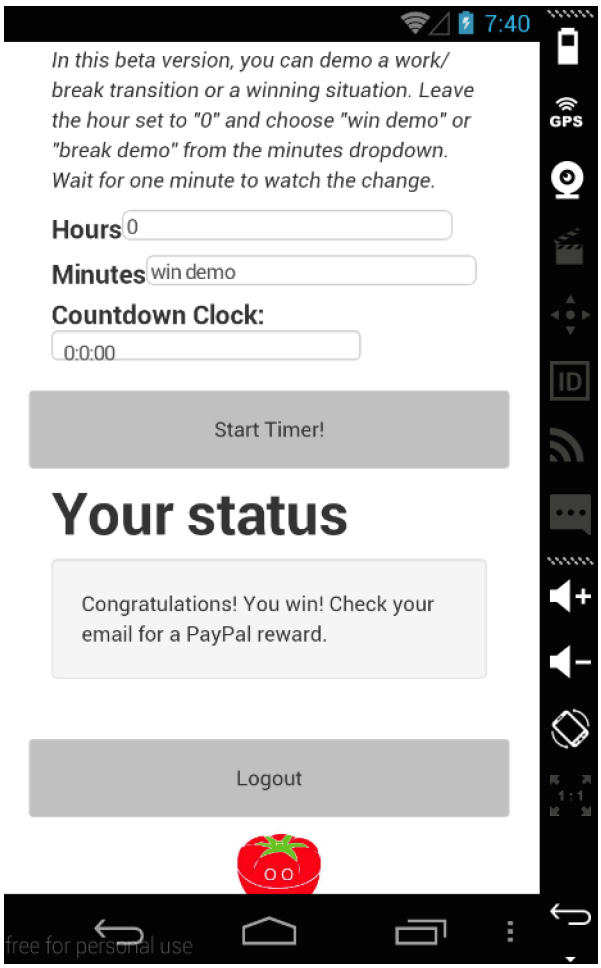
22. Select the number of hours from the “Hours” dropdown menu.

	
23. Select the number of minutes from the “Minutes” dropdown menu. The “win demo” sets the timer for 1 minute, so the user can demo a win. The “break demo” sets the timer for 6 minutes, so the user can demo a transition between a focus period and a break period.	24. Break demo, focus period. The status is “Time to focus. You can do it!”

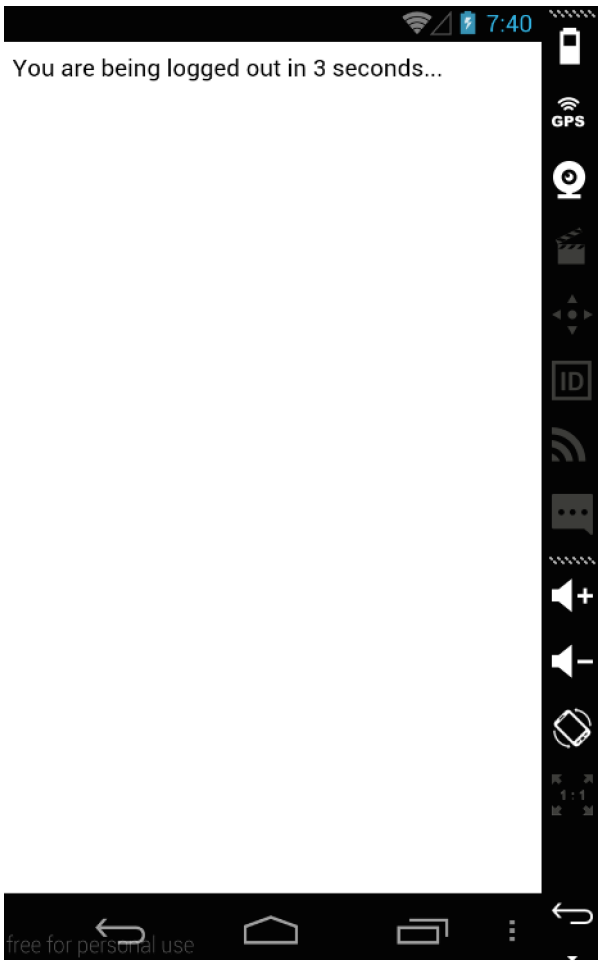
	
25. Navigation warning 1. If the user clicks the back button to leave the app, they will be warned.	26. Navigation warning 2. A second warning pops up after the first.



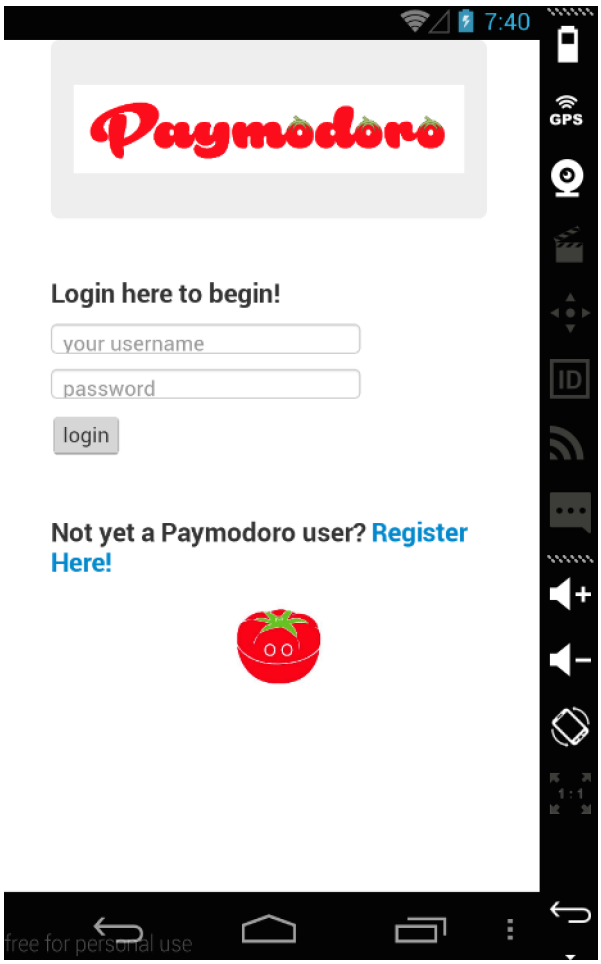
27. Message from a break period. Notice the status is “Whew… take a break. Great job!”



28. Winning message. Notice the congratulatory message under “Your Status.”

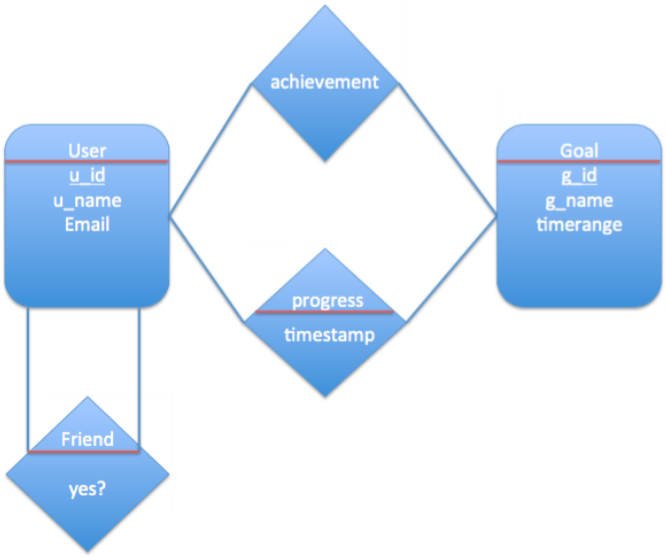


29. When the user logs out from the timer screen, they see this redirect message.



30. After logging out, the user is redirected to the login screen.

Appendix Part 3: Table Designs



SQL code:

```
create table User(
u_id int(60) AUTO_INCREMENT primary key,
u_name varchar(60),
password varchar(60),
Email varchar(60),
honor int(60))
create table Goal(
g_id int(60) AUTO_INCREMENT primary key,
g_name varchar(60),
time_range int(60),
tags varchar(60))
create table Progress(
u_id int(60),
g_id int(60),
timestart timestamp,
primary key (u_id, g_id),
foreign key (u_id) references User(u_id),
foreign key (g_id) references Goal(g_id))
create table Achievement(
u_id int(60),
g_id int(60),
primary key (u_id, g_id),
foreign key (u_id) references User(u_id),
foreign key (g_id) references Goal(g_id))
```

```
create table Friend(
from_user int(60),
to_user int(60),
yes int(1),
primary key (from_user, to_user),
foreign key (from_user) references User(u_id),
foreign key (to_user) references Goal(u_id))
```

User table contains u_id ,u_name, Email information, which contains the basic information of the Users. The u_name is used mainly to keep the unchangeable information, for we will edit our profile where we can change our Email. So we store and identify the user by u_name.

Future versions of Paymodoro will include user-defined goals, and also the ability to "friend" other Paymodoro users.

Goal table will store users' goals; we will allow users to choose from a limited set of predefined goals. This part can be used in update, as we will introduce new goals with every new version.

Achievements are pretty similar to Progress, except Progress contains more attributes in the date. They mainly contain the u_id and g_id to associate the connection with user and goal. For Progress, this table will contain goals that users are currently pursuing, and the time when the users create the goals. The achievements contain all the goals the user has finished. We will determine later whether we want to display all the achievements one specific user has.

Friends table contains two friends and a flag whether a user agrees or not to another user's friend request. After a user sends a request to another, a tuple will be added in the friends table, with from_user(the requesting user), to_user(the response user), yes="0"(means that the request has been sent but have not agreed by the other user). However when another user agrees, then it will updated to the "yes" in the tuple as "1", meaning that they are friends. However, when a user refuses the user who requests, we will just delete the tuple.