A Project Report on

Habitio: Habit Tracker & Productive App Using Hybrid Model

Submitted in partial fulfilment of the

Requirement for the award of the degree of

B.Tech in Computer Science & Engg.

GALGOTIAS UNIVERSITY

(Established under Galgotias University Uttar Pradesh Act No. 14 of 2011)

Under The Supervision of

Dr. A. John

(Associate Professor)

Submitted By

Puskar Bharti

19SCSE1010664

SCHOOL OF COMPUTING SCIENCE AND ENGINEERING

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

GALGOTIAS UNIVERSITY, GREATER NOIDA

INDIA October, 2021

DECLARATION

I hereby declare that the project entitled “Habitio” submitted for the B.Tech in Computer Science and Engineering is my original work and the project has not formed the basis for the award of any other degree, B.Tech, fellowship or any other similar titles.

Place: Signature

Date: (Puskar Bharti)

CERTIFICATE

This is to certify that the project titled **”Habitio”** is the bona fide work carried out by **Puskar Bharti** student of B tech in computer science and Engineering , Galgotias University , Gautam buddh Nagar, Uttar Pradesh (India) during the academic year 2019-22, In partial fulfilment of the requirements for the award of the B tech in computer science and Engineering and that the project has not formed the basis for the award previously of any other degree, B tech, fellowship or any other similar title.

Place: Guide Signature

Date: (Dr. A. John)

TABLE OF CONTENTS

ACKNOWLEDGEMENT

This document lays out a project plan for the development of the “Habitio” an open source Habit Tracker & Productive App Using Hybrid Model system.

The intended readers of this document is developer working on “Habitio” and the guide of the project. The plan will include, but is not restricted to, a summary of the system functionality, the scope of the project from the perspective of the “Habitio” team (me and my mentors), scheduling and delivery estimates, project risks, and how those risks will be mitigated, the process by which I will develop the project, and metrics and measurements that will be recorded throughout the project.

I owe my profound and sincere gratitude to my project supervisor and guide

Dr. A. John, who showed avid interest and guided me throughout in my project work that has been titled —”Habitio: Habit Tracker & Productive App Using Hybrid Model” till the culmination of my project by giving us all the requisite information for establishing the project. The project advancement aided us in my research and I were able to know and find out many new things in this domain.

ABSTRACT

It is well established that lack of physical activity is detrimental to the overall health of an individual. Modern-day activity trackers enable individuals to monitor their daily activities to meet and maintain targets. This is expected to promote activity encouraging behaviour, but the benefits of activity trackers attenuate over time due to waning adherence. One of the key approaches to improving adherence to goals is to motivate individuals to improve on their historic performance metrics.

The aim of this work was to build an android app to predict an achievable weekly activity target by considering patterns in the user's activity tracker data in the previous week and behaviour and environment characteristics. By setting realistic goals, ones that are neither too easy nor too difficult to achieve, activity tracker users can be encouraged to continue to meet these goals, and at the same time, to find utility in their activity tracker.

Habit tracking is powerful for three reasons.

1. It creates a visual cue that can remind us to act.

2. It is motivating to see the progress you are making. We don't want to break our streak.
3. It feels satisfying to record our success in the moment.

Following would be the technologies I would use to develop my application within the stipulated time period:

Front-end development: React Native, JavaScript.

Back-end development: NodeJS, JavaScript, SQL.

Database: MongoDB.

1. INTRODUCTION

Habit tracking is the process of measuring whether we repeat a particular action. Every day we complete our task, our habit streak is extended. The more streak we have, the more motivation we get to try to prolong our streak. The most basic format is to get a calendar and cross off each day we stick with our routine. For example, if we meditate on Monday, Wednesday, and Friday, each of those dates gets an X. As time rolls by, the calendar becomes a record of our habit streak. Habit tracking naturally builds a series of visual cues. When we look at the calendar and see our streak, we'll be reminded to act again.

Research has shown that people who track their progress on goals like losing weight, quitting smoking, and lowering blood pressure are all more likely to improve than those who don't. One study of more than sixteen hundred people found that those who kept a daily food log lost twice as much weight as those who did not. A habit tracker is a simple way to log your behaviour, and the mere act of tracking a behaviour can spark the urge to change it.

Habit tracking also keeps us honest. Most of us think we act better than we do. Measurement offers one way to overcome our blindness to our own behaviour and

notice what's really going on each day. When the evidence is right in front of you, you're less likely to lie to yourself.

1.1 Problem Statement

One of the most common mistakes people make when they try to stick with high-performance habits is not measuring their progress properly. That's why they ultimately fail to stay on track.

Most of us jump into the whole habit forming process without wondering about the habit of tracking our progress regularly.

If we want to make our tracking more efficient, we need to do it consistently and regularly. In other words, we should make it habitual. Once we do it in such an automatic and natural way, we will put it into the muscle memory of your brain. Then, we do it as an automatic and natural way. And if we wish to improve ourselves, we need to understand where we're at first.

That's why we always need to look back at our progress, and the best way to assess our performance is through statistics.

1.2 Objective

Habit metrics are really powerful when it comes to self-reflection. It allows us to go beyond the surface of whether we're doing the habit or not, and gain deeper self-understanding, define factors affecting our behaviours, hence deriving realistic ways to improve ourselves. But all things becomes a bit overwhelming and sound a bit complicated.

And to perform all those things in a simple yet effective way I have designed an app "Habitio", which enables people to track their habit and set goals to increase their productivity and feel satisfactory and motivated.

2. LITERATURE SURVEY

2.1 Study of existing systems and feasibility of project

Habits play an important role in supporting behaviour change and ensuring it has long-term effects. Once a person makes a decision to change their behaviour and takes action, that action needs to be regularly repeated. To ensure the change becomes permanent, the repetition needs to be maintained until the task becomes automatic; the new behaviour needs to turn into a habit.

Elements of Habit Formation

Habits are defined as automatic responses to contextual cues (e.g. location, existing routine events, objects or pre ceding actions). They form as the behaviour is repeated in a stable context and the repetition helps to create associations between the task and its cues. Behaviour can be considered automatic when it reaches the automaticity plateau i.e. the asymptote of a curve representing the relationship between repetition and habit strength. The number of repetitions

required to reach the asymptote depends on the complexity of the task and it can vary from 18 days for easy tasks (e.g. drinking more water) to an estimated 254 days for more complex tasks (e.g. going to the gym). However, repetition alone is not enough to form a habit.

Cues and trigger events support the habit formation process, as they start to drive the behaviour. Existing routines can be used as prompts to action, as tasks linked to routine events (event-based tasks), e.g. taking medication after breakfast, are generally easier to remember than tasks that need to be completed at a specified time (time-based tasks), e.g. meditating at 10pm every evening. Although associations between the task and contextual cues form automatically through repetition, it is possible to steer this process by forming implementation intentions. Implementation intentions are action plans in the following format: "When situation X arises, I will perform response Y", e.g. "when I finish eating dinner, I will drink a glass of water". They help to connect the new behaviour with an existing routine and turn it into an event-based task. When the relationship between the task and its cues is explicitly stated, each repetition reinforces that association, which leads to a more efficient action initiation in the future and increases the automaticity of the behaviour. The trigger routine needs to be relevant and

meaningful to make it easier to associate it with the new task, and needs to be reliable, i.e. occur as frequently as is desired of the new target behaviour.

External memory aids (e.g. reminders, notes) can also serve as cues and play an important role in supporting habit development. They are especially useful when they refer to the target behaviour and the situation in which it needs to be executed, although the effectiveness and salience of reminders decreases with time. While automatically responding to a reminder could be seen as a habit, it is not related to the target behaviour and does not help to make that behaviour automatic. People who expect to be reminded score worse in prospective memory tests, as they put less mental effort into trying to remember and therefore are more likely to forget. However, in some cases reminders could support the start of a new habit, as the automaticity of the new behaviour might develop faster than the decay of effectiveness of the reminder. Another factor that can influence habit formation is positive reinforcement. Even small successes increase the feeling of satisfaction and can strengthen the habit. Satisfaction can also trigger the feeling of being in control, which reinforces the need to repeat the action in the future. These feelings help with maintaining long-term behaviour change, as they increase the belief that starting the new behaviour was a good choice. Therefore, to

successfully form a habit, people need to start identifying the execution of the task with its rewarding nature.

Rewards can be extrinsic, such as financial incentives, or intrinsic, such as pleasure or satisfaction. However, there is a danger that if they are extrinsic and expected, they will hinder habit formation by reducing intrinsic motivation. While extrinsic rewards can still help to develop automaticity of the behaviour, they may not be feasible or practical and it might be difficult to distinguish between whether the action is truly habitual and whether people are engaged just to get the reward. Thus, extrinsic rewards are likely to facilitate habit formation only when the reward is not a goal in itself and the behaviour offers other, ideally intrinsic, benefits to the person. However, people develop habits even when they do not receive any explicit positive reinforcement, which suggests that while it can support habit formation, reinforcement plays a lesser role in the process than other factors.

The increasing popularity of smartphones (58% of US adults owned one in 2014) makes them an ideal plat form for delivering targeted, low cost interventions. Thousands of behaviour change apps are available in various app stores and are being used as tools for behaviour change. However, as we discuss in the next section, they hardly ever incorporate features that support habit formation.

CHANGING BEHAVIOR WITH APPS

Behaviour change apps tend to focus on personal health and wellness, physical activity and healthy eating, although they could help with other types of behaviours, such as good work habits, e.g. making to-do lists every morning or house chores, e.g. washing up after meals. As any type of behaviour targeted by these apps requires regular repetition and can be associated with relevant contextual cues, it could benefit from habit support. However, behaviour change apps often do not support habit formation, which is partly related to the fact that they tend not to be grounded in research.

STUDY: REVIEW OF HABIT FORMATION APPS

Hundreds of habit formation apps are currently available and can be downloaded with a single click. Results of our previous study and existing design recommendations for designing apps that support users' daily routines suggest that effective apps should allow users to select trigger events that would serve as cues

and help maintain repetition until a habit is formed. However, habit formation apps have not been evaluated by academic researchers and their effectiveness or theoretical grounding are not known. Therefore, we conducted a study to investigate what functionality they offer. As habit formation is part of a broader behaviour change process, the study explored whether the apps were grounded in both habit formation and behaviour change research.

Method

The keyword "habit" was used to search for habit formation apps in the UK version of Apple iTunes Store (http://www.apple.com/itunes/) and Google Play (https://play.google. com/store/apps). The search was conducted in April 2014 and returned 859 apps (553 for iPhones and 306 for Android phones). Results were scanned to identify apps designed specifically to support the development of new habits. The following types of apps were excluded as they did not support habit formation: habit cessation apps, general behaviour change apps, food and activity trackers, and exercise routines, books about habits, and research apps that require registration codes. Apps for tablets and apps not available in English were also excluded. In the end, 115 apps were selected as relevant: 54 Android apps and 67

iPhone apps. Six apps were available for both platforms, but since they had identical descriptions, they were counted only once.

A list of feature categories was created based on descriptions of 20 identified apps (10 from each app store). App features were listed in detail and grouped into 14 broader feature categories. These categories were later used in the main data collection phase and for each of the 115 identified apps their presence was noted. Supporting features, such as backups, data export or password protection, were also noted, but were excluded from the analysis as they were not directly related to habit formation.

To assess whether the apps support habit formation, we coded for whether the features of the apps supported the use of contextual cues, helped to form implementation intentions or provided positive reinforcement. The assessment was done. Next, since habit formation is a part of the wider behaviour change process, features were also matched with corresponding behaviour change techniques from Behaviour Change Techniques Taxonomy. Based on the Taxonomy, a list of techniques that could be delivered by smartphone apps was created. Items from the list were then matched with functionality by the first author. To validate the results, matching techniques were presented as a list to two

other researchers who were asked to independently match them with functionality by selecting up to three techniques that could be supported by each feature.

Result

The functionality review showed that apps primarily focused on providing features that support self-tracking; they did not seem to be designed to explicitly support habit formation. Self-monitoring is important in the early stages of the behaviour change process and is often used in interventions as it helps people understand their behaviour, set realistic goals, monitor progress and maintain motivation. However, it does not help them form associations between the task and the environment, nor does it support the development of automaticity. Using the app to track their own behaviour may help users see trends, but there is a danger that it might also teach them to depend on technology. Presence of reminders also teaches users to rely on them, as does positive reinforcement. Because of this dependence, apps that require constant engagement might hinder the development of automaticity of behaviour.

Only five out of 14 identified feature categories could be matched with factors supporting habit formation and only one of them - routine creation - could help

users to find the right trigger event. At the same time, all features could be matched with behaviour change techniques, which is encouraging, but also suggests a lack of understanding of habit formation and its role in supporting behaviour change.

3. SYSTEM DEVELOPMENT

The most basic format is to get a calendar and cross off each day you stick with your routine. For example, if you meditate on Monday, Wednesday, and Friday, each of those dates gets an X. As time rolls by, the calendar becomes a record of your habit streak.

To make this process as easy as possible, I created the Habit Journal, which includes 12 habit tracker templates—one for each month. All you have to do is add your habit and start crossing off the days.

Placing an X on each day is the classic look. I prefer something a little more design-oriented, so I shade in the cells on my habit tracker. You could also use checkmarks or fill your habit tracker with dots.

No matter what design you choose, the key point is your habit tracker provides immediate evidence that you completed your habit. It's a signal that you are making progress. Of course, that's not all it does… Habit tracking is powerful for three reasons.

1. It creates a visual cue that can remind you to act.
2. It is motivating to see the progress you are making. You don't want to break your streak.
3. It feels satisfying to record your success in the moment.

Let's break down each one.

Benefit #1: A habit tracker reminds you to act.

Habit tracking naturally builds a series of visual cues. When you look at the calendar and see your streak, you'll be reminded to act again.

Research has shown that people who track their progress on goals like losing weight, quitting smoking, and lowering blood pressure are all more likely to

improve than those who don’t. One study of more than sixteen hundred people found that those who kept a daily food log lost twice as much weight as those who did not. A habit tracker is a simple way to log your behavior, and the mere act of tracking a behavior can spark the urge to change it.

Habit tracking also keeps you honest. Most of us think we act better than we do. Measurement offers one way to overcome our blindness to our own behavior and notice what’s really going on each day. When the evidence is right in front of you, you’re less likely to lie to yourself.

Benefit #2: A habit tracker motivates you to continue.

The most effective form of motivation is progress. When we get a signal that we are moving forward, we become more motivated to continue down that path. In this way, habit tracking can have an addictive effect on motivation. Each small win feeds your desire.

This can be particularly powerful on a bad day. When you’re feeling down, it’s easy to forget about all the progress you have already made. Habit tracking provides visual proof of your hard work—a subtle reminder of how far you've come. Plus, the empty square you see each morning can motivate you to get started because you don't want to lose your progress by breaking your streak.

Benefit #3: A habit tracker provides immediate satisfaction.

Finally, tracking feels rewarding. It is satisfying to cross an item off your to-do list, to complete an entry in your workout log, or to mark an X on the calendar. It feels good to watch your results grow and if it feels good, then you're more likely to endure.

Habit tracking also helps keep your eye on the ball: you're focused on the process rather than the result. You're not fixated on getting six-pack abs, you're just trying to keep the streak alive and become the type of person who doesn't miss workouts.

Habit Tracker Ideas

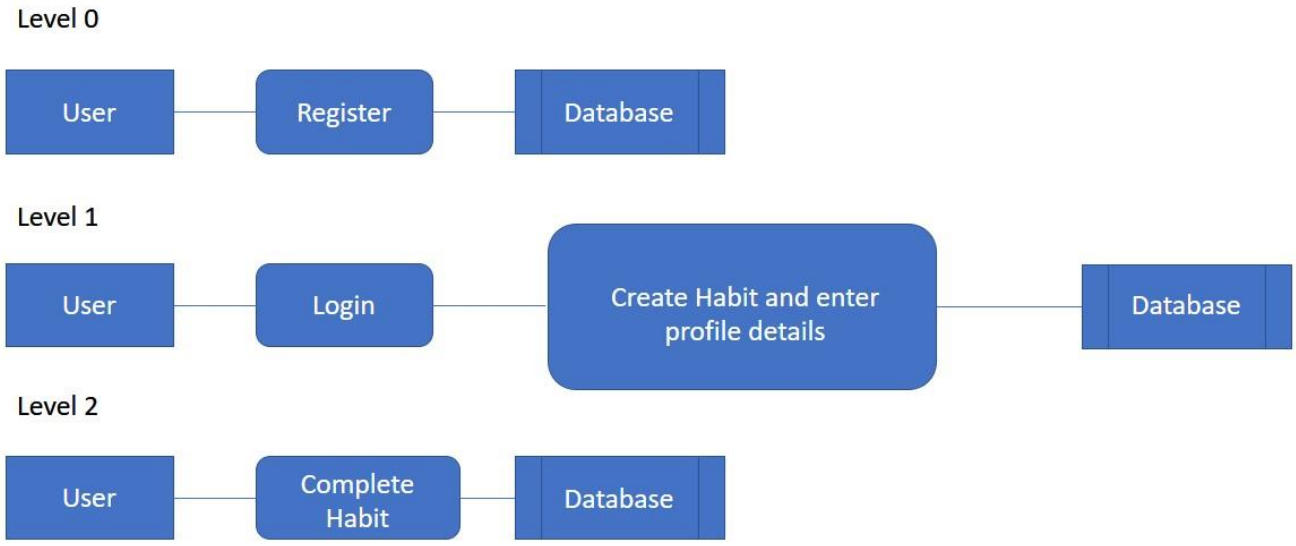
Alright, those benefits sound great, but it's not necessary to fill your habit tracker with every habit that makes up your day. In fact, if you're already sticking to a habit, then it seems like extra work to me to track it as well. So what should you measure in your habit tracker?

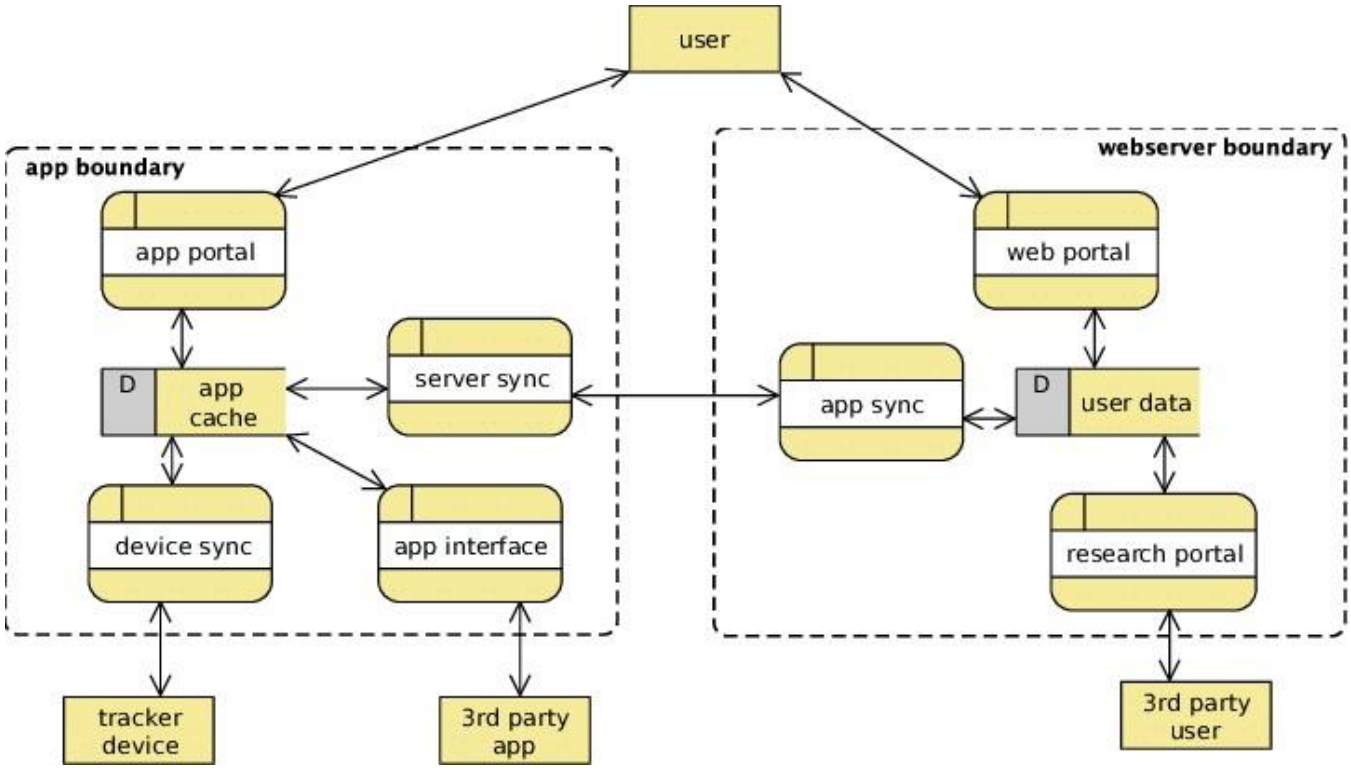
Habit tracking can help kick-start a new habit or keep you on track with behaviours that you tend to forget or let slide when things get busy.

In Atomic Habits, I recommend using the Two-Minute Rule, which suggests you scale your habits down until they take two minutes or less to perform. You can track whatever habits you want in your habit tracker, but I recommend starting with

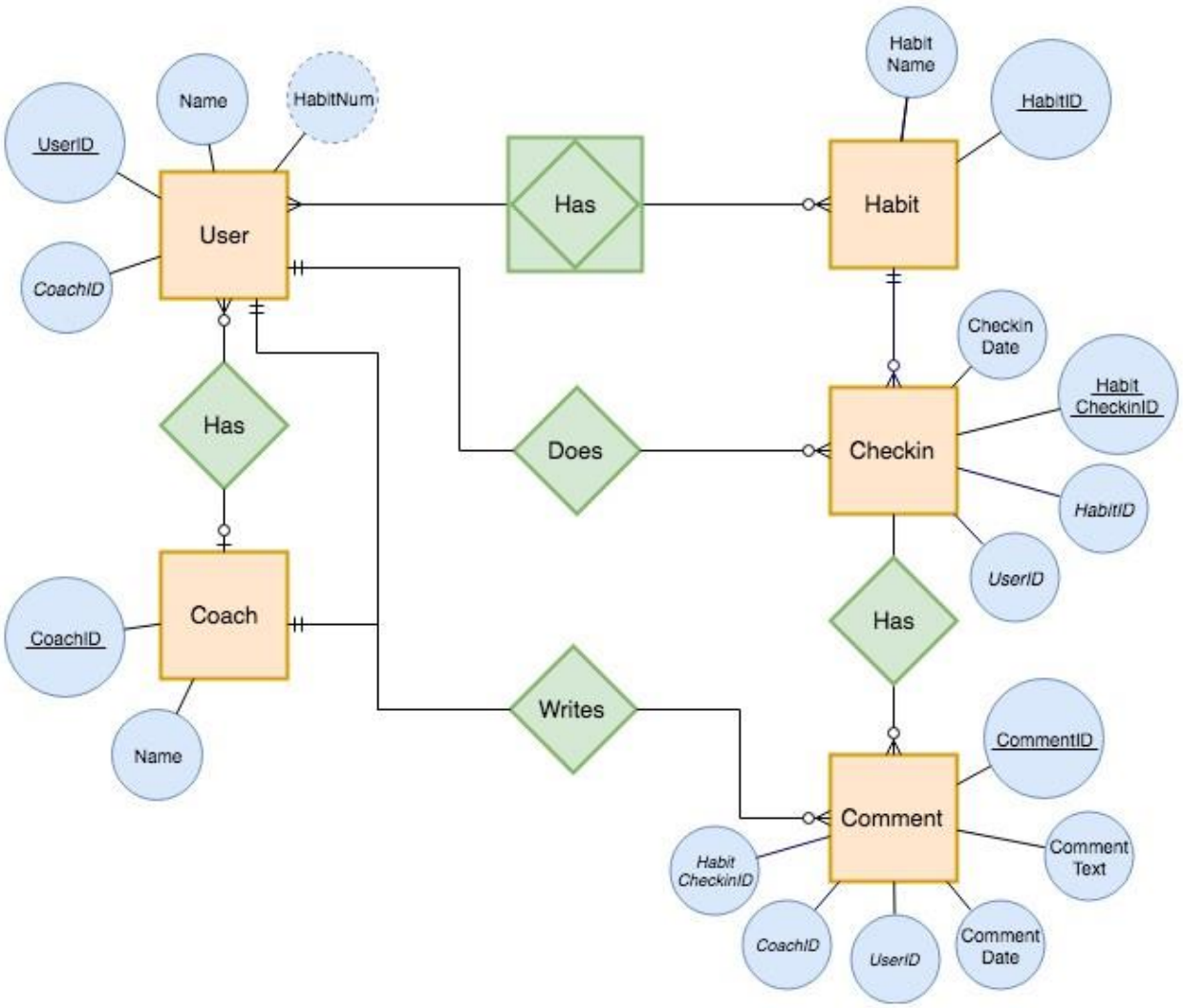
these super small habits to make sure that you are at least showing up in a small way each day.

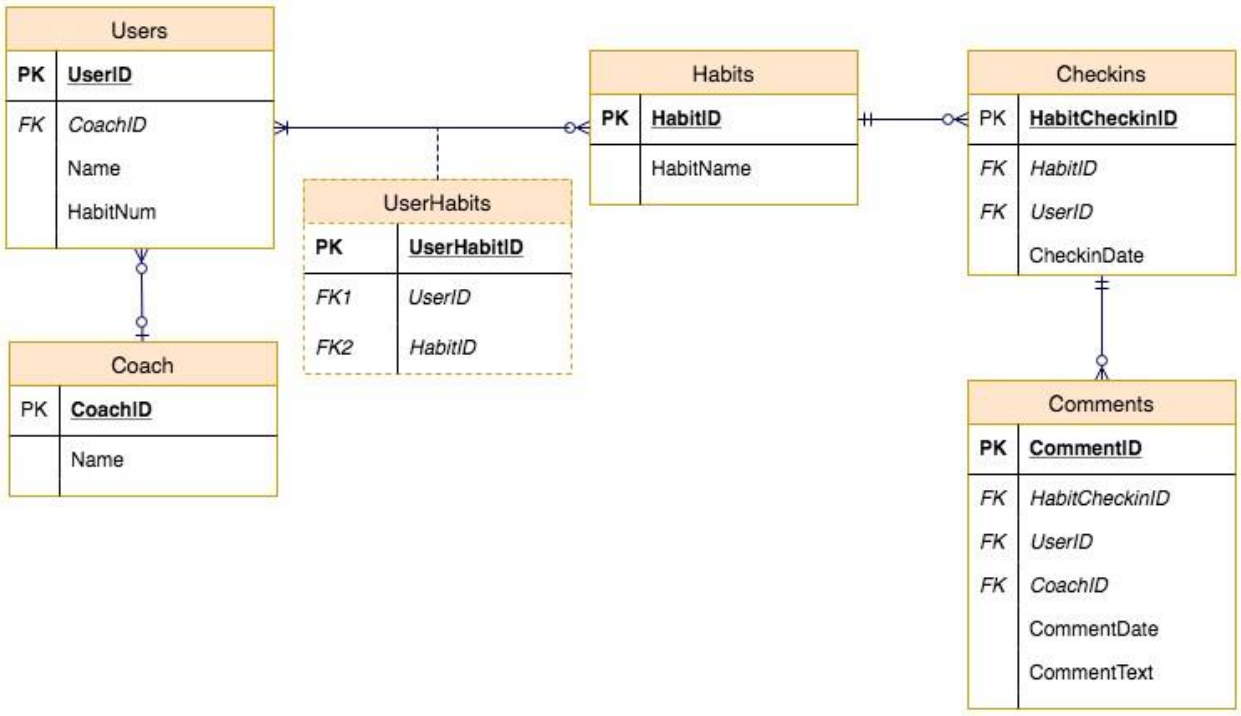
DFD – Data Flow Diagram




user
app boundary
app portal
D
app cache
server sync
device sync
app interface
tracker device
3rd party app
webserver boundary
web portal
app sync
D
user data
research portal
3rd party user

ER Diagram


Habit Name
HabitID
Name
HabitNum
UserID
Has
Habit
User
CoachID
Checkin Date
Habit CheckinID
Has
Does
Checkin
HabitID
UserID
Coach
CoachID
Has
Writes
Name
CommentID
Comment
Habit CheckinID
Comment Text
CoachID
UserID
Comment Date



Design Guidelines for Habit Formation

In this section, we provide explicit guidance for the design of future habit formation apps

Support trigger events Allow users to form implementation intentions and explicitly ask them to select trigger events, eg. "I will do X after eating breakfast" (for more information on how this could be done). Monitor their behavior by asking later if the task was completed. If users keep forgetting, suggest selecting a different trigger event.

Use reminders to reinforce implementation intentions Re mind users of their implementation intentions in advance by sending notifications before their selected trigger actions, eg. "Please remember to do Y after brushing your teeth" of "Don't forget to do Z before going to sleep. This could help users form associations between the task and its trigger, and would encourage them to remember on their own. To ensure users do not become reliant on notifications, they should phase out with time.

Avoid features that teach users to rely on technology. Re minders and selftracking teach users to rely on the technological solution and cues interfere with the process of developing associations between contextual cues and the task. They should not be used in habit formation apps as they hinder the process of habit formation

Pseudo Code

➢ SQL Code for Table Creation

```
CREATE TABLE MyGuests (
id INT(6) UNSIGNED AUTO_INCREMENT PRIMARY KEY,
firstname VARCHAR(30) NOT NULL,
lastname VARCHAR(30) NOT NULL,
email VARCHAR(50),
reg_date TIMESTAMP DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP
)
```

➢ PHP Code for Database Connectivity

Example (MySQLi Object-Oriented)

```php
<?php
$servername = "localhost";
$username = "username";
$password = "password";

// Create connection
$conn = new mysqli($servername, $username, $password);

// Check connection
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}
echo "Connected successfully";
?>
```

➢ PHP Code for Inserting Data in Table

Example (MySQLi Object-oriented)

```php
<?php
$servername = "localhost";
$username = "username";
$password = "password";
$dbname = "myDB";

// Create connection
$conn = new mysqli($servername, $username, $password, $dbname);
// Check connection
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}

$sql = "INSERT INTO MyGuests (firstname, lastname, email)
VALUES ('John', 'Doe', 'john@example.com')";

if ($conn->query($sql) === TRUE) {
    echo "New record created successfully";
} else {
    echo "Error: " . $sql . "<br>" . $conn->error;
}

$conn->close();
?>
```

➢ PHP Code for Updating Data in Table

Example (MySQLi Object-oriented)

```php
<?php
$servername = "localhost";
$username = "username";
$password = "password";
$dbname = "myDB";

// Create connection
$conn = new mysqli($servername, $username, $password, $dbname);
// Check connection
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}

$sql = "UPDATE MyGuests SET lastname='Doe' WHERE id=2";

if ($conn->query($sql) === TRUE) {
    echo "Record updated successfully";
} else {
    echo "Error updating record: " . $conn->error;
}

$conn->close();
?>
```

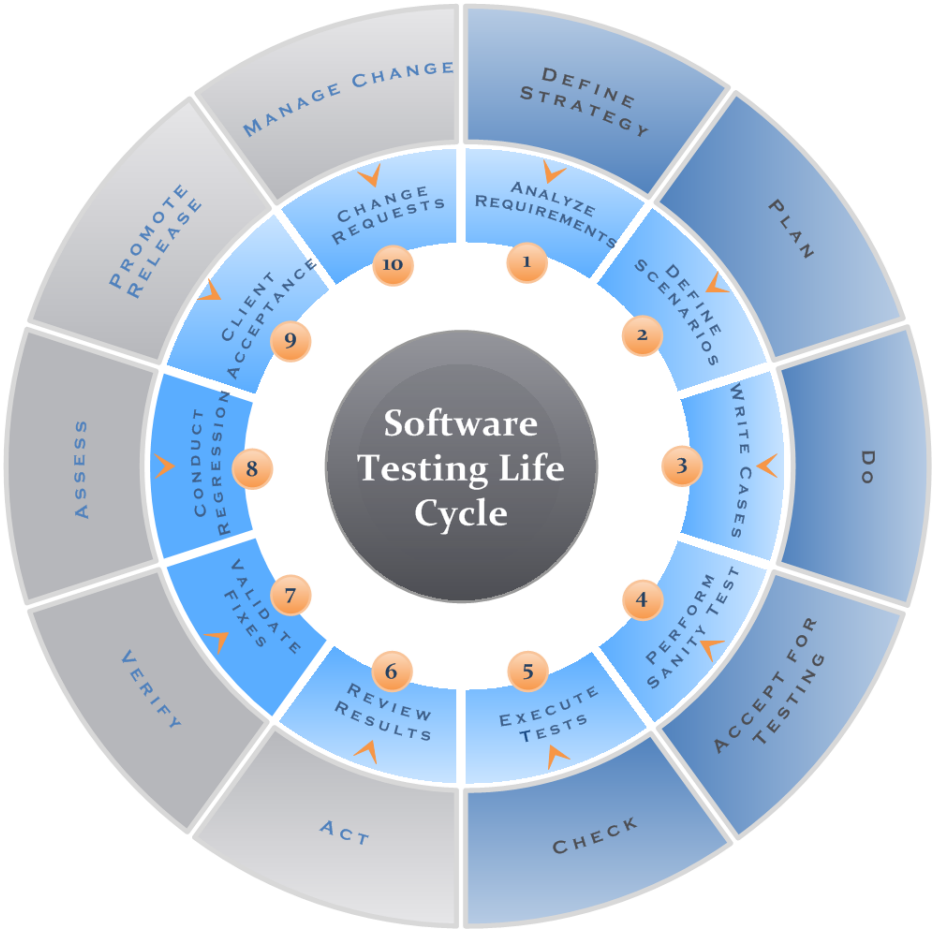
➢ PHP Code for Deleting Data from Table

Example (MySQLi Object-oriented)

```php
<?php
$servername = "localhost";
$username = "username";
$password = "password";
$dbname = "myDB";

// Create connection
$conn = new mysqli($servername, $username, $password, $dbname);
// Check connection
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}

// sql to delete a record
$sql = "DELETE FROM MyGuests WHERE id=3";

if ($conn->query($sql) === TRUE) {
    echo "Record deleted successfully";
} else {
    echo "Error deleting record: " . $conn->error;
}

$conn->close();
?>
```

4. System Testing

The internet defines Software Testing as the process of executing a program or application with the intent of identifying bugs. I like to define Testing as the process of validating that a piece of software meets its business and technical requirements. Testing is the primary avenue to check that the built product meets requirements adequately.



1. **Test Strategy and Test Plan**

Every project needs a Test Strategy and a Test Plan. These artefact describe the scope for testing for a project:

- The systems that need to be tested, and any specific configurations
- Features and functions that are the focus of the project
- Non-functional requirements
- Test approach—traditional, exploratory, automation, etc.—or a mix
- Key processes to follow – for defects resolution, defects triage
- Tools—for logging defects, for test case scripting, for trace-ability
- Documentation to refer, and to produce as output
- Test environment requirements and setup
- Risks, dependencies and contingencies
- Test Schedule
- Approval work flows
- Entry/Exit criteria

And so on… Whatever methodology your project follows, you need to have a Test Strategy and Software Testing Plan in place. Make them two separate documents, or merge them into one.

Without a clear test strategy and a detailed test plan, even agile projects will find it difficult to be productive. Why, you ask? Well, the act of creating a strategy and plan bring out a number of dependencies that you may not think of otherwise.

For example, if you're building a mobile app, a test strategy will help you articulate what Operating Systems (iOS/Android), OS versions (iOS 7 onwards, Android 4.4 onwards etc.), devices (last three generations of each type of iOS device, specific models of Android devices) you need to test the app for.

Usually, a Functioning Organization Will Have Nailed Their Device And OS Support Strategy, *And Review It Quarterly* to keep up with the market; test managers creating a strategy or plan for their project will help validate the enterprise-wide strategy against project-specific deliverable.

You'd be surprised how many projects have to alter their plan significantly because they hadn't thought enough about support strategy early on. Among other things, the test plan also helps define **entry** and **exit criteria** for testing. This is important as a control for the rest of the team. If the deliverable aren't of a specific level of

quality, they won't enter testing; similarly, if the tested code doesn't meet specific

Quality standards, the code will not move to the next phase or enter production.

Testing performs this all-important gate-keeping function, and *Helps Bring Visibility* to any issues that may be brushed under the carpet otherwise.

2. Test Design

Now that you have a strategy and a plan, the next step is to dive into creating a test suite. A test suite is a collection of test cases that are necessary to validate the system being built, against its original requirements.

Test design as a process is an amalgamation of the Test Manager's experience of similar projects over the years, testers' knowledge of the system/functionality being tested and prevailing practices in testing at any given point. For instance, if you work for a company in the early stages of a new product development, your focus will be on uncovering major bugs with the alpha/beta versions of your software, and less on making the software completely bug-proof.

The product may not yet have hit the critical “star” or “cash cow” stages of its existence—it’s still a question mark. And you probably have investors backing you, or another product of your own that is subsiding this new initiative until it can break even. Here, you’re trying to make significant strides—more like giant leaps—with your product before you’re happy to unwrap it in front of customers. Therefore, you’re less worried about superficial aspects like look and feel, and more worried about fundamental functionality that sets your product apart from your competitors.

In such a scenario, you may use lesser negative testing and more exploratory or disruptive testing to weed out complex, critical bugs. And you may want to leave out the more rigorous testing to until you have a viable product in hand. So your test suite at the beginning of the product life cycle will be tuned towards testing fundamentals until you’re close to release.

When you are happy to release a version to your customers, you’ll want to employ more scientific testing to make it as bug-free as possible to improve customer experience. On the other hand, if you’re testing an established product or system, then you probably already have a stable

test suite. You then review the core test suite against individual project requirements to identify any gaps that need additional test cases.

3. **Test Execution**

You can execute tests in many different ways—as single, waterfall SIT (System Integration Test) and UAT (User Acceptance Test) phases; as part of Agile sprints; supplemented with exploratory tests; or with test-driven development. Ultimately, you need to do adequate amount of software testing to ensure your system is (relatively) bug-free.

Let's set methodology aside for a second, and focus on how you can clock adequate testing. Let's go back to the example of building a mobile app that can be supported across operating systems, OS versions, devices. The most important question that will guide your test efforts is "what is my test environment?".

You need to understand your test environment requirements clearly to be able to decide your testing strategy. For instance, does your app depend on integration with a core system back end to display information

and notifications to customers? If yes, your test environment needs to provide back end integration to support meaningful functional tests.

Can you commission such an end-to-end environment to be built and ready for your sprints to begin? Depending on how your IT organisation is set up, maybe not. This is where the question of agile vs a more flexible approach comes into picture. Could you have foreseen this necessity way before the sprints began? Probably not.

Given how Agile projects are run, you may only have a couple of weeks between initiating a project and starting delivery sprints, which time isn't enough to commission an end-to-end test environment if one doesn't already exist. If everything goes fine, you'll have a test environment to your liking, configured to support your project, with all enablers built to specifications. If not, then your test strategy will be different.

In this example, we're talking about doing front-end tests with dummy back end to support in-sprint testing, and wait until an integrated test environment is ready. It is common practice to schedule integration tests just after delivery sprints and before release. Your team can then run a dedicated System Integration Test, focusing on how the app components work with the back end to deliver the required functionality. So while

app-specific bugs will primarily be reported during the sprints, functional end-to-end bugs will crop up during the integration test. You can follow this up with a UAT cycle to put finishing touches in terms of look and feel, copy, etc. How your team execute test cycles depends on the enabling infrastructure, project and team structure in your organization.

Reviewing Test Environment Requirements Early On Is Now A Widely Recognized Cornerstone For Good Project Management. Leaders are giving permanent, duplicate test environments a good deal of thought as an enabler for delivery at pace.

4. Test Closure

Right—so you have done the planning necessary, executed tests and now want to green-light your product for release. You need to consider the exit criteria for signal-ling completion of the test cycle and readiness for a release. Let's look at the components of exit criteria in general:

- 100% requirements coverage: all business and technical requirements have to be covered by testing.

- Minimum % pass rate: targeting 90% of all test cases to be passed is best practice.
- All critical defects to be fixed: self-explanatory. They are critical for a reason.

As a rule of thumb, I've seen projects mandate 90% pass rate and all critical defects being fixed before the team can move on to the next phase of the project. And on big transformation initiatives, I've seen individual releases move to the next phase (to aid beta pilots) with as little as 80%, with the understanding that the product won't reach the customer until mandatory exit criteria are met. Ultimately, what works for your team is down to your circumstances and business demands.

Remember that nobody can afford serious defects to remain unfixed when you launch to customers—especially if your product handles sensitive information or financial.

Polish things off with a Test Summary and Defects analysis: providing stats about testing – how many high/medium/low defects, which functions/features were affected, where were defects concentrated the most, approaches used to resolve defects (defer vs fix), Traceability Matrix to demonstrate requirements coverage.

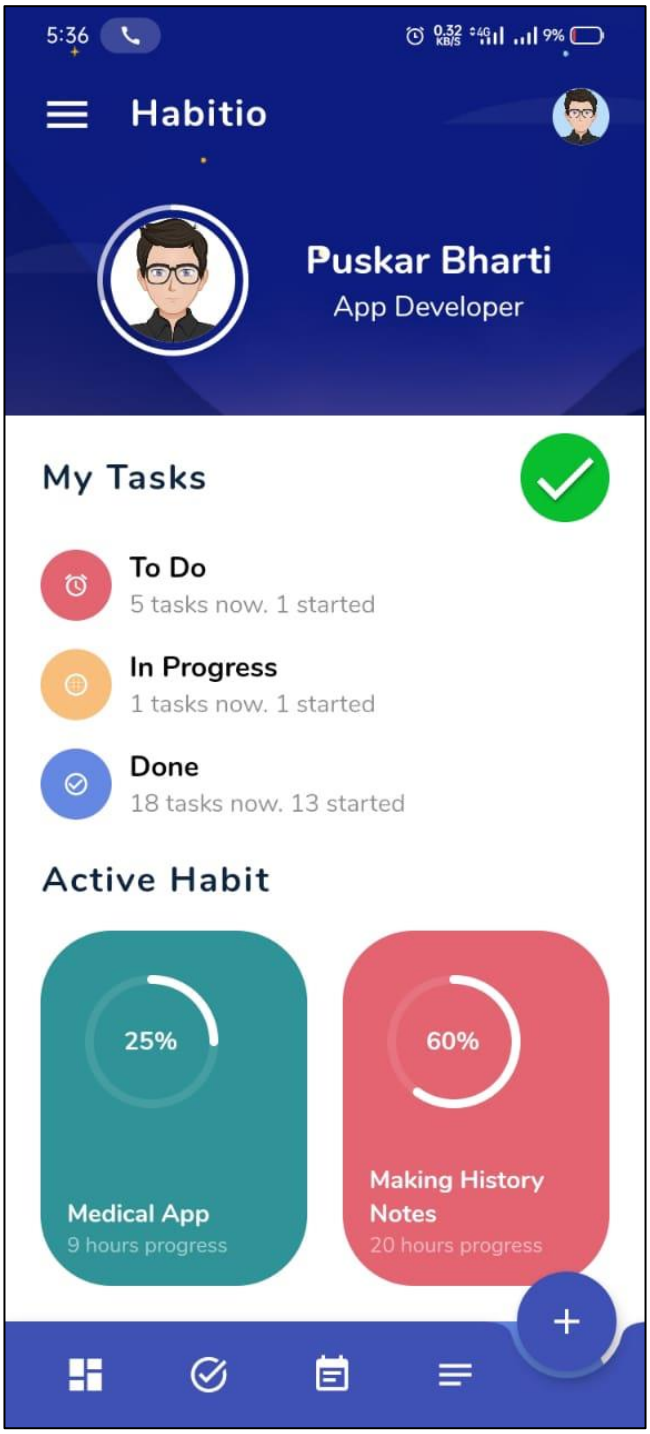
5. LIMITATIONS AND FUTURE WORK

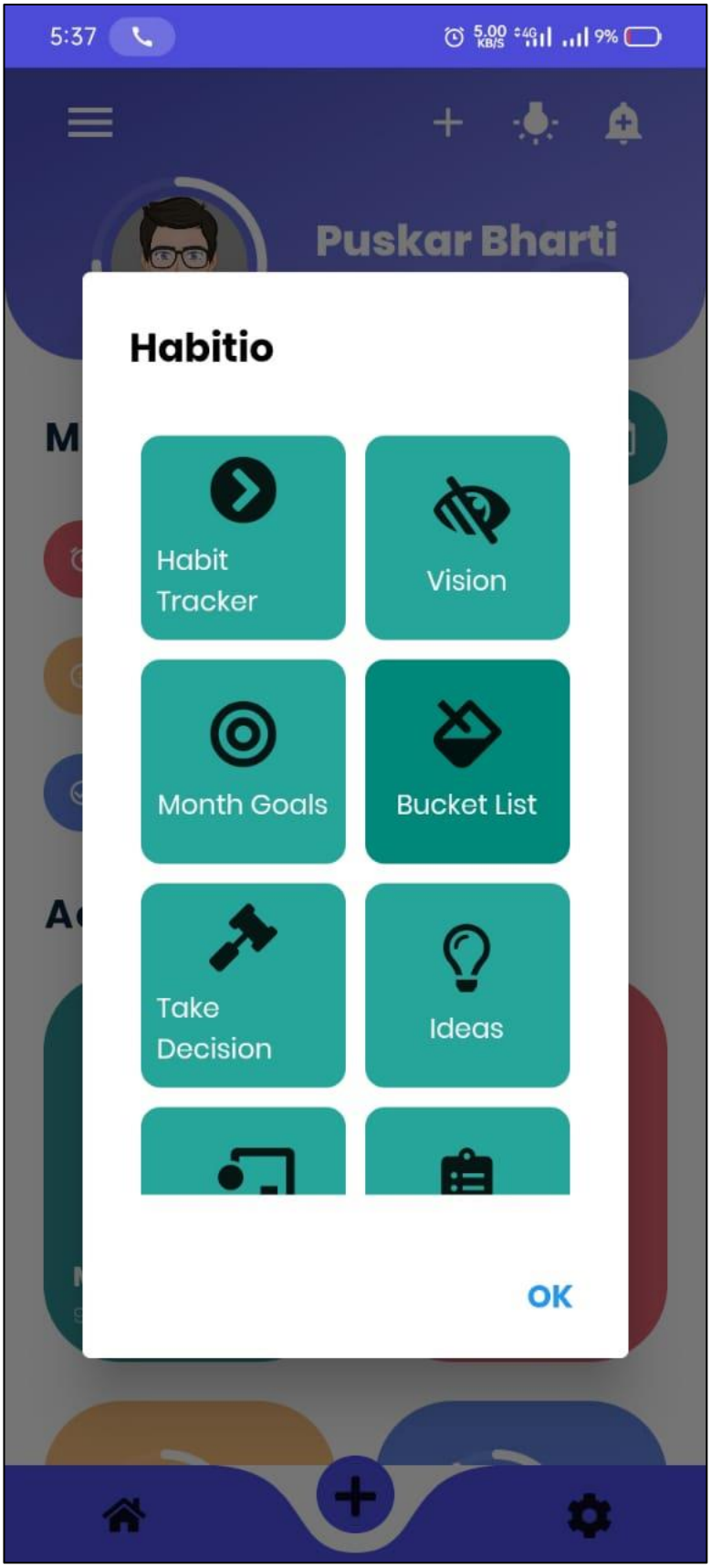
On average, it takes 66 days to form a habit. However, simple tasks become automatic quicker than complex actions. We assumed that sending an SMS was simple enough to observe increase in automaticity in only four weeks. While our assumption turned out to be connect, future work in this area should validate the results over longer periods

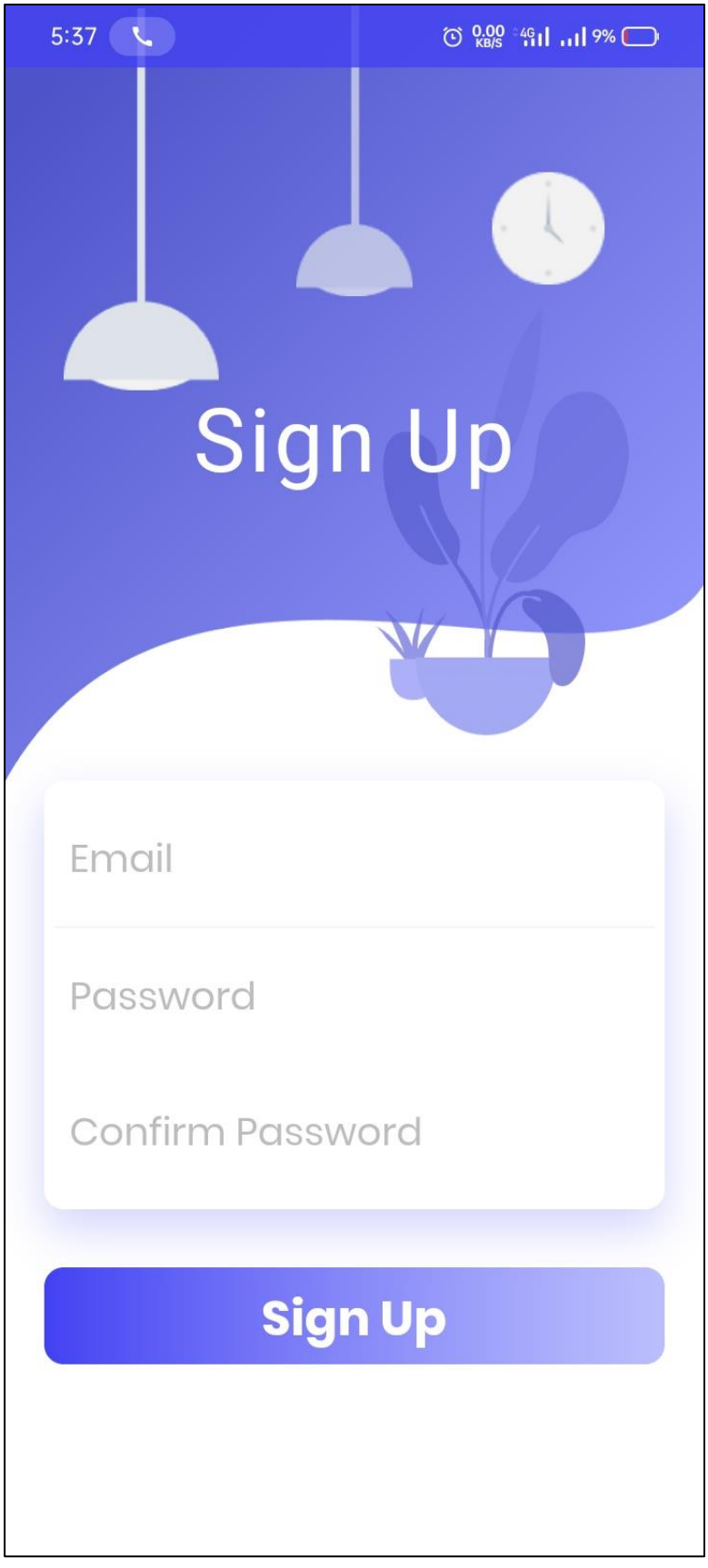
The positive reinforcement we provided was not effective. Possibly due to the way it was delivered or its content Technical issues experienced during the first two weeks of the study might also have had an impact by reducing the efficacy of our interventions. We decided to analyse the fall study data because most participants received most of the messages Removing participants who had not received all text messages from the data set would have made it difficult to compare positive reinforcement conditions with others. It would also require limiting data from the whole study to just two weeks, which is too short a period to measure the impact on habit formation Moreover, data would have been confounded by the fact that the study had already been running for two weeks. While habits can form without

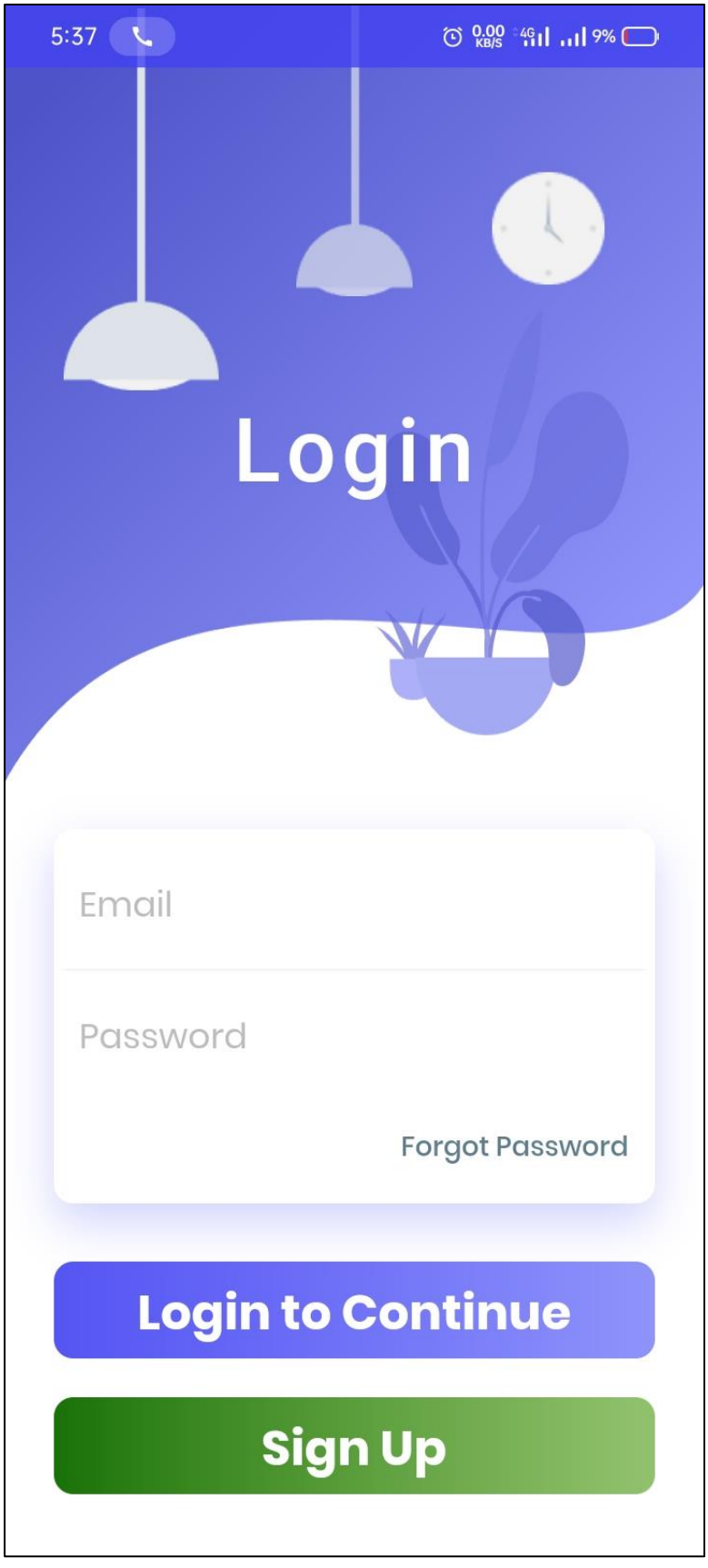
explicit positive reinforcement, understanding its links with intrinsic rewards and the development of automaticity requires further studies.

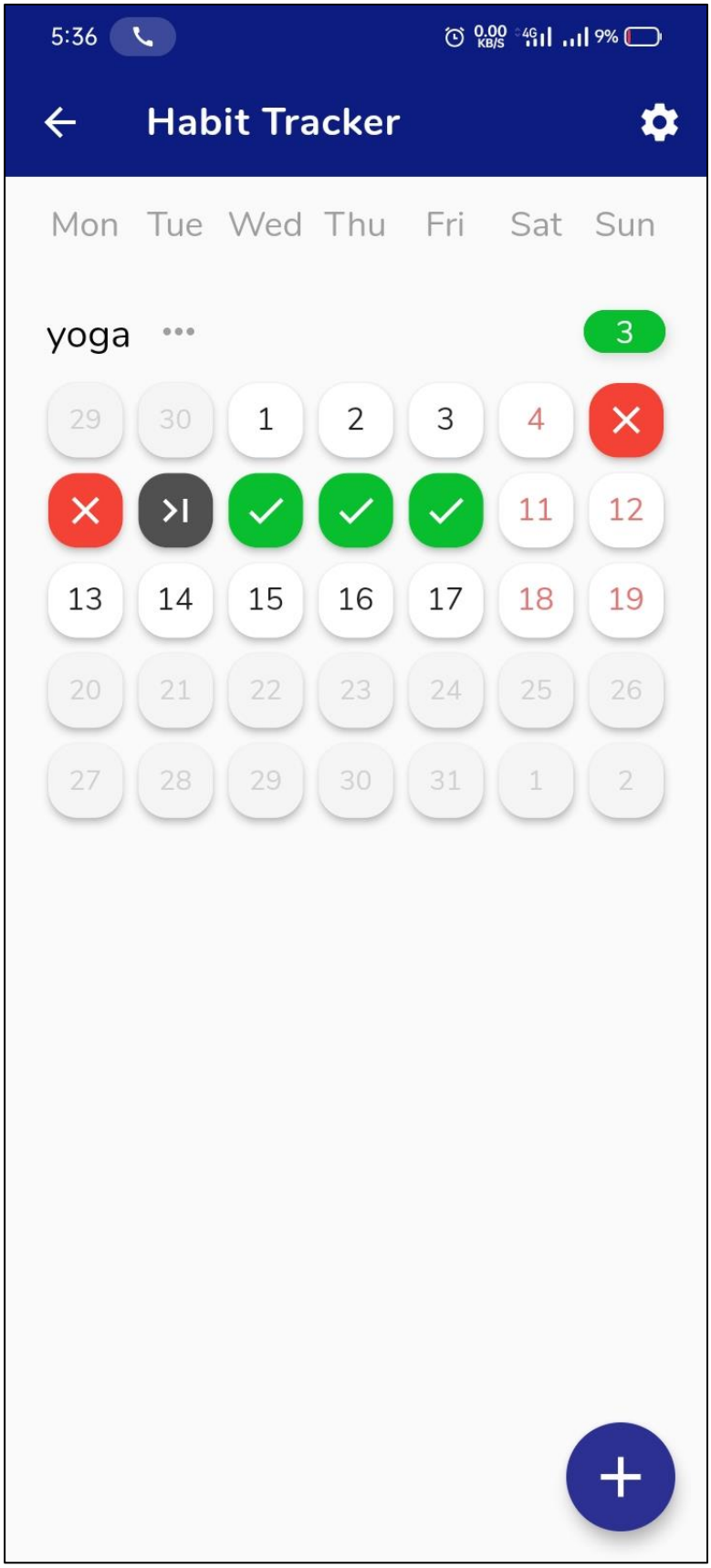
Screenshot – Result

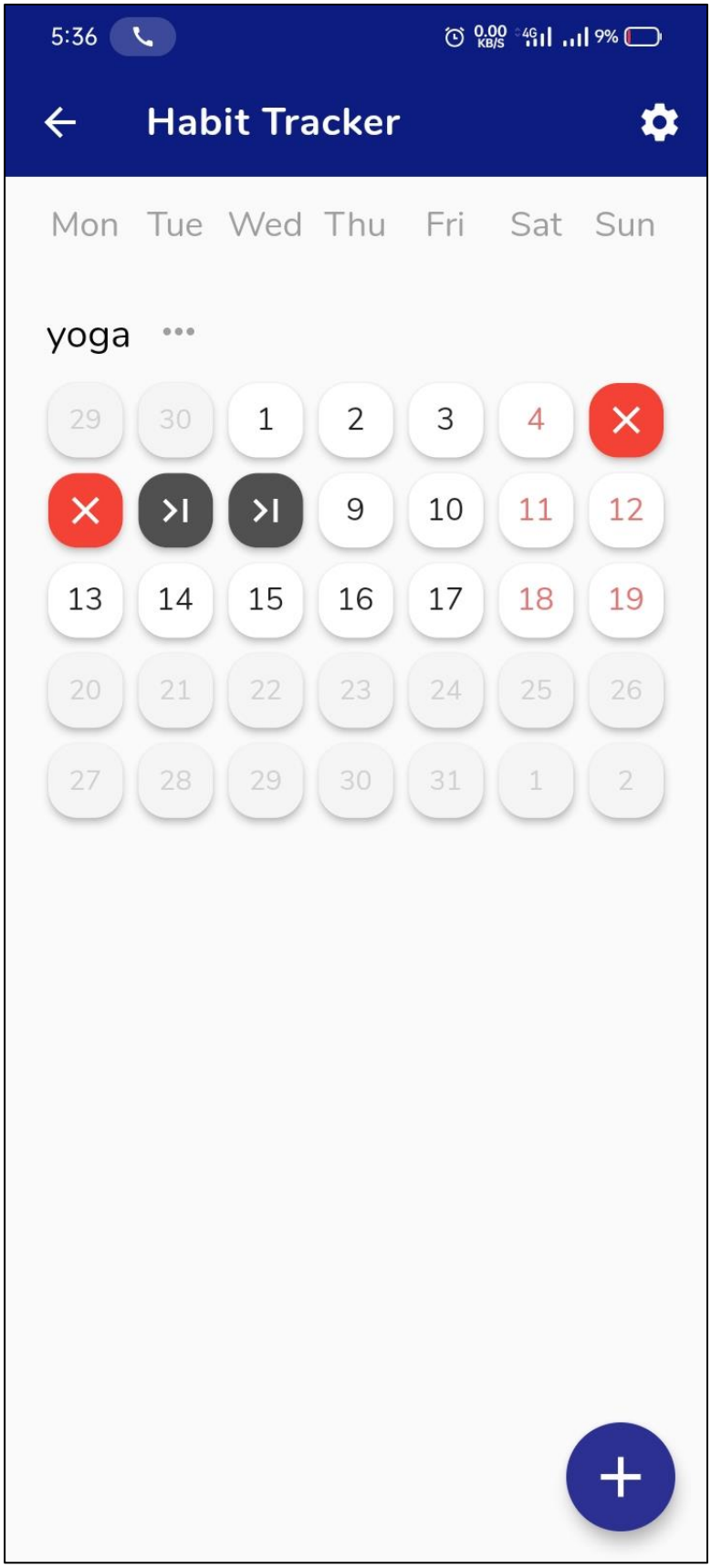


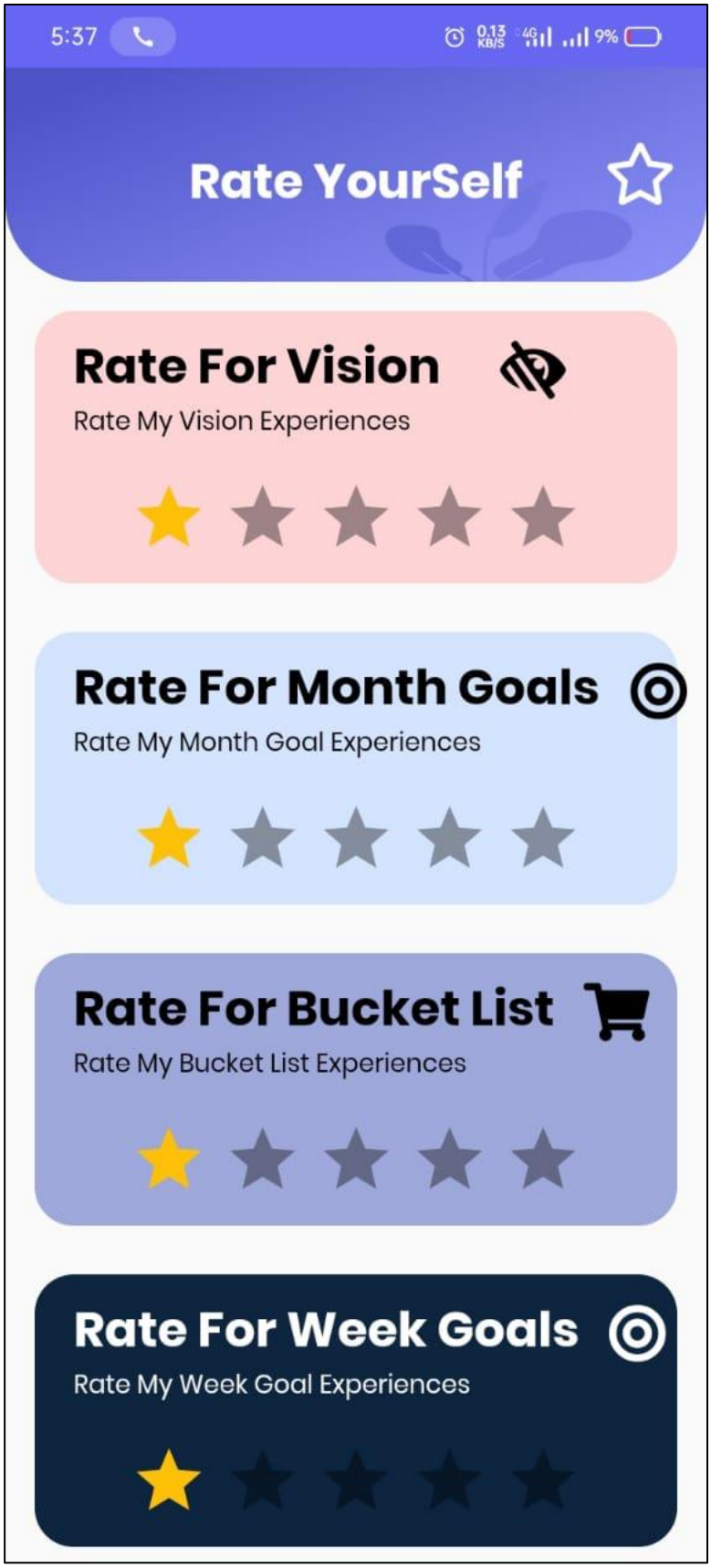

5:37
Puskar Bharti
Habitio
Habit Tracker
Vision
Month Goals
Bucket List
Take Decision
Ideas
OK

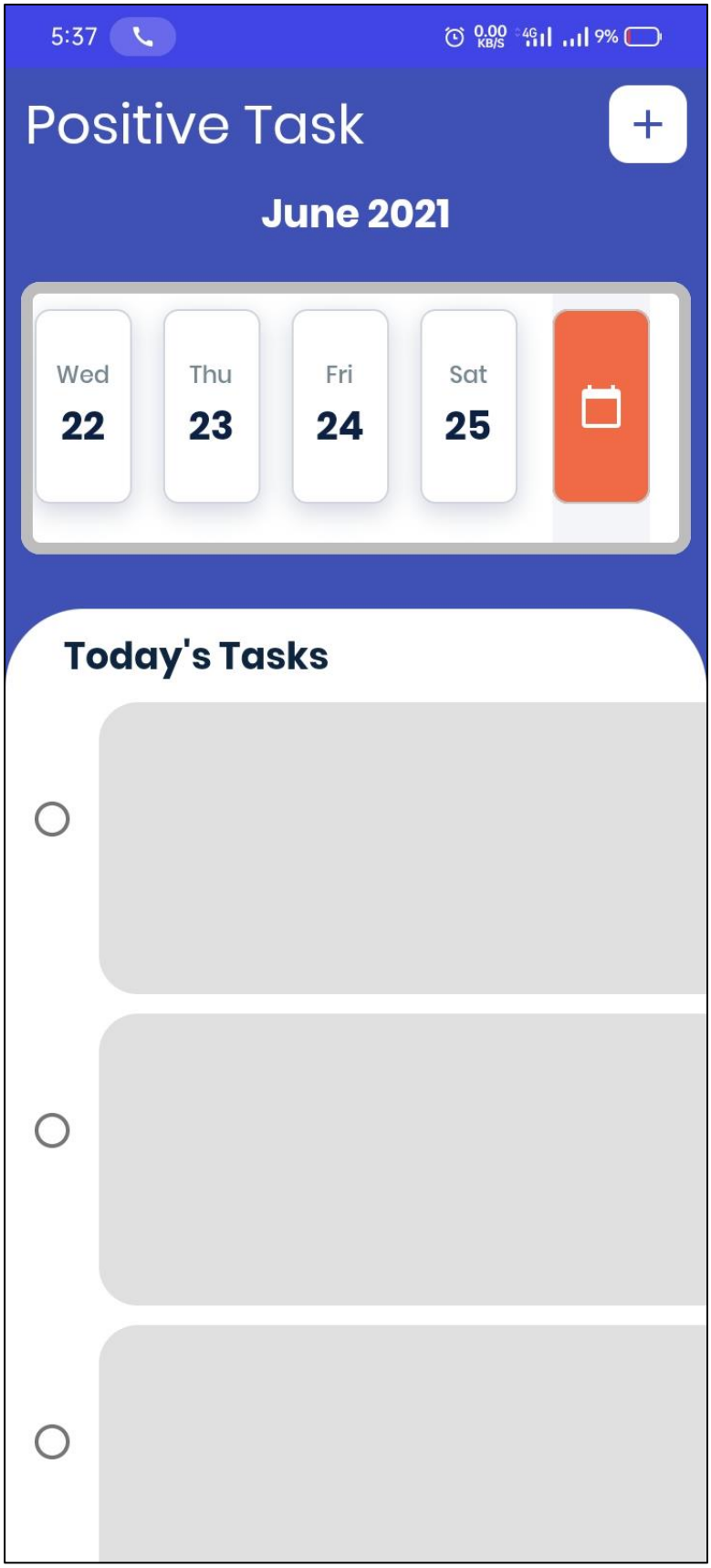

5:37
0.00 KB/S
9%
Sign Up
Email
Password
Confirm Password
Sign Up

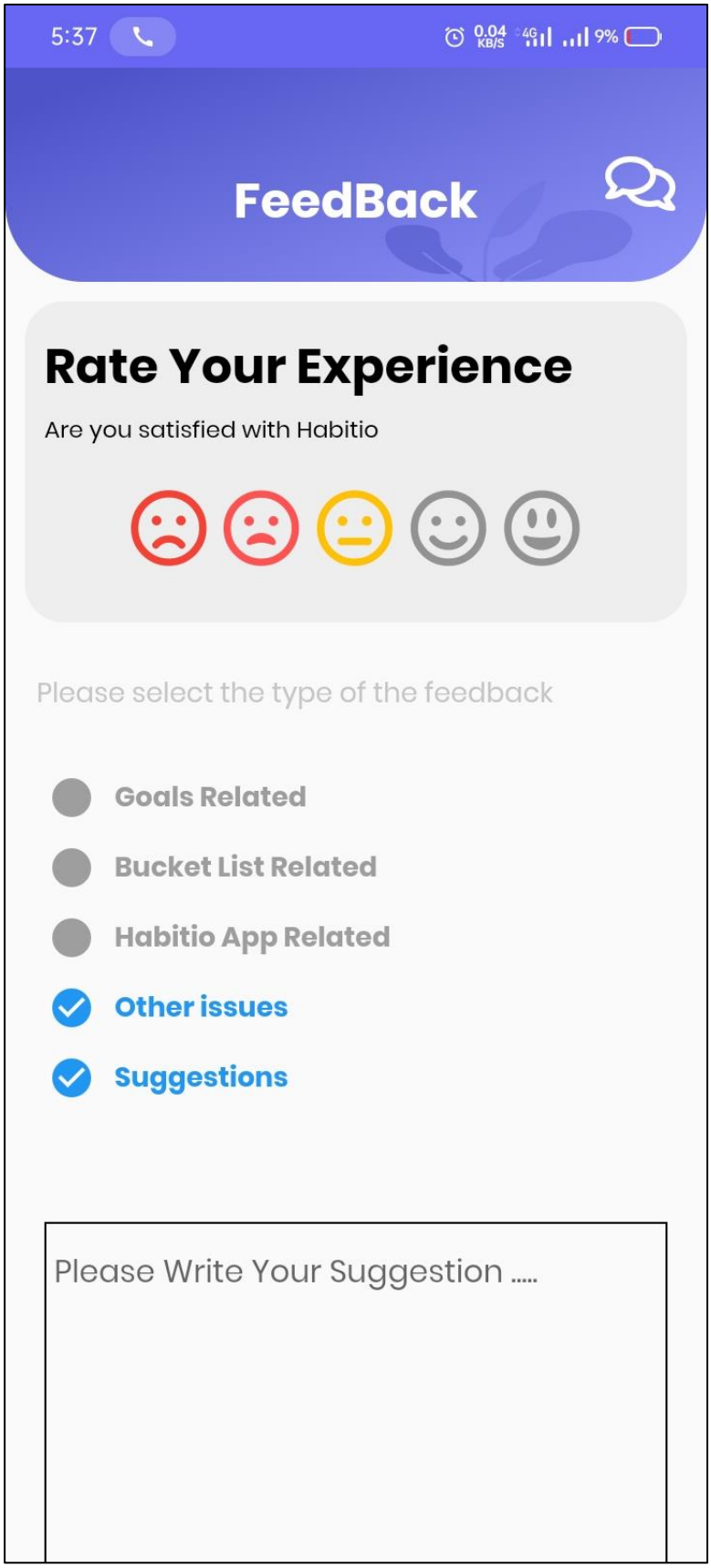

5:37
0.00 KB/S
9%
Login
Email
Password
Forgot Password
Login to Continue
Sign Up

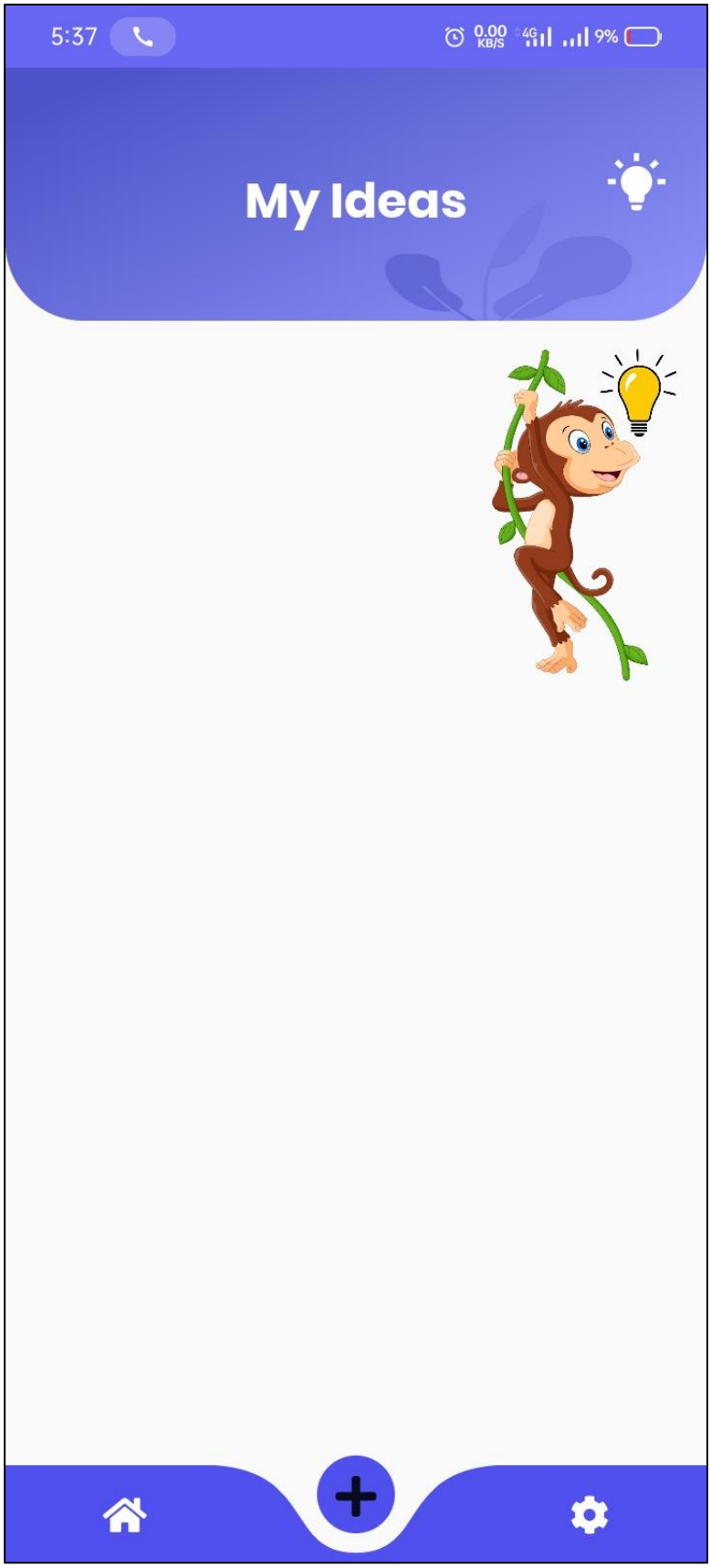

5:36
0.00 KB/S
9%
Habit Tracker
Mon Tue Wed Thu Fri Sat Sun
yoga
3
29 30 1 2 3 4
11 12
13 14 15 16 17 18 19
20 21 22 23 24 25 26
27 28 29 30 31 1 2

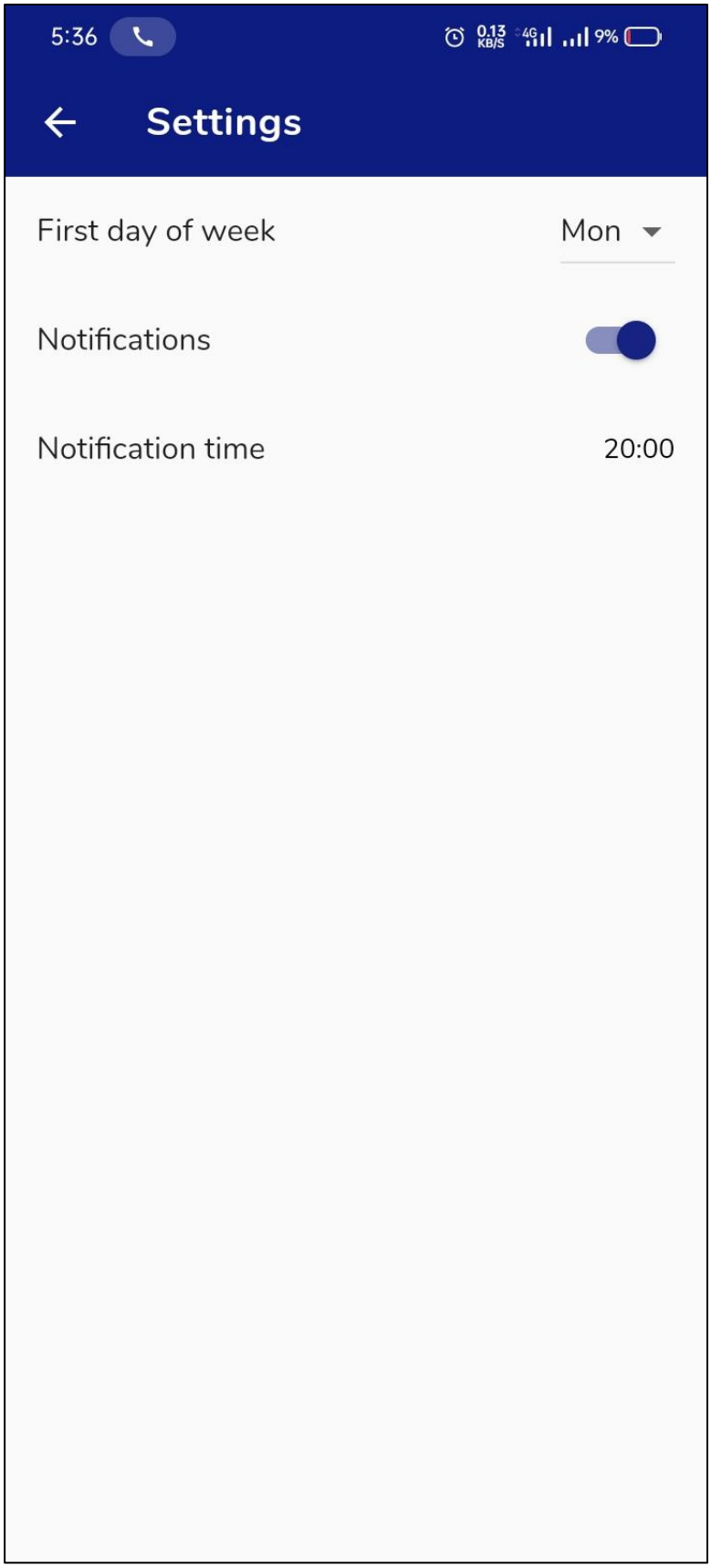

5:36
0.00 KB/S
9%
Habit Tracker
Mon Tue Wed Thu Fri Sat Sun
yoga
29 30 1 2 3 4
9 10 11 12
13 14 15 16 17 18 19
20 21 22 23 24 25 26
27 28 29 30 31 1 2


5:37
9%
Rate YourSelf
Rate For Vision
Rate My Vision Experiences
Rate For Month Goals
Rate My Month Goal Experiences
Rate For Bucket List
Rate My Bucket List Experiences
Rate For Week Goals
Rate My Week Goal Experiences


5:37
0.00 KB/S
9%
Positive Task
+
June 2021
Wed
22
Thu
23
Fri
24
Sat
25
Today's Tasks


5:37
FeedBack
Rate Your Experience
Are you satisfied with Habitio
Please select the type of the feedback
Goals Related
Bucket List Related
Habitio App Related
Other issues
Suggestions
Please Write Your Suggestion .....


5:37
0.00 KB/S
9%
My Ideas


5:36
9%
Settings
First day of week
Mon
Notifications
Notification time
20:00

CONCLUSIONS

This paper makes three contributions that are of interest to the CHI community.

First, we show how reminders and trigger events influence habit formation.

Secondly, we highlight the fact that currently available habit formation

Apps are not grounded in the habit literature and do not help users associate their new behaviour with trigger events. Finally, we present design guidelines for technology-based behaviour change interventions that support habit formation

Developing automaticity of a new behaviour can help to ensure that the change will have long-lasting results. As habits are an important part of behaviour change, we need to better understand how the mechanisms of habit formation can be facilitated by mobile technologies. We have presented the results of two studies exploring how technology could support habit formation and outlined design guidelines that can help us move away from self-tracking and reminders. By supporting contextual cues and implementation intentions, apps and other behavior change technologies could help users develop new long-lasting habits.

REFERENCES

Aarts, H Paulussen, T und Schaalma, H Physical exercise babit: on the conceptualization and formation of habitual health behaviours Health Education Research, 12,3(1997), 363-374

2Bandura, A. Social cognitive theory of self-regulation Organizational Behavior and Human Decision Processes 50, 2(1991), 248-287.

BCT Taxonomy. (2014) Behaviour Change l'echnique

Taxonomy, butp://www.bet-taxonomy.com/

Consolvo, S. Klasnja, P., McDonald, DW... Avrahami, D. Froehlich, J. LeGrand L Landay, J.A. Flowers or a robot army Encouraging awareness & activity with personal, mobile displays. In Proc. UbiComp 2008, ACM Press (2008), 54-63