Training at

QBURST TECHNOLOGIES, KORATTY

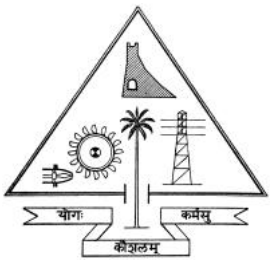
UG INDUSTRIAL TRAINING REPORT

MAY, 2017

Prepared by

AMRUTHA M U

ETANECS008



Department of Computer Science & Engineering
Government Engineering College

Thrissur-680009

Contents

List of Figures

List of Tables

Nomenclature

ADT	Android Development Tools
API	Application Program Interface
GUI	Graphical User Interface
HTTP	Hyper Text Markup Language
IDE	Integrated Development Environment
REST	Representational State Transfer
SCM	Source Code Management
SDK	Software Development Kit
SNS	Social Networking Sites
URL	Uniform Resource Locator
UUID	Universally Unique Identifier
WWW	World Wide Web
XML	eXtensible Markup Language

1 Introduction

1.1 Scope of the training

The Industrial Training was undergone at QBurst Technologies, Koratty during the period of 20th June, 2016 to 5th August, 2016 (Excluding Weekends). 14 interns from selected from Government Engineering College, Thrissur with a stipend of Rs. 10000 per month. The internship selection program consisted of a coding test (C Language) and 20 students were shortlisted. The shortlisted students had to go through an interview which contained questions to test programming skills, web development background, managerial skills etc. At the end of the day, 14 students were selected as the 2016-2017 batch interns of QBurst Technologies, Koratty.

The four areas of Computer Science that QBurst technologies focuses on are Web Development, Mobile App Development, Big Data Analytics and Cloud Computing. The selected 14 interns were divided into 3 groups. One group was assigned projects on Android , the second one on iOS and the third group was given the responsibility of back-end development.

1.2 Organization of report

This report is organized into seven chapters - Introduction, About the Industry, Training Schedule, Life at the Industry, Topics Covered, Software Development Tools and Conclusion. The company profile and history is detailed in the chapter About the Industry. Complete schedule of the 45 day training programme is tabulated in the Training Schedule chapter. There is also an elaborate chapter on Life at the Industry. Detailed training procedure and topics covered is explained in the chapter, Topics Covered. All the technologies and tools used during the training are enlisted in the chapter, Software Development Tools.

2 About the Industry

QBurst Technologies was founded in 2004. It has got a well defined mission and a vision. Its mission is to be the worlds best and most sought after firm for custom web and mobile application development. And its vision is to provide their clients with reliable, superior software at a cost-effective price while making the development process an enjoyable experience for clients as well as employees.

The areas of expertise of QBurst Technologies are: Web, Mobile, Cloud, Design, Validation and Testing, BI and Analytics, Web and Social Intelligence etc. Some of the technologies employed at QBurst Technologies are: HTML 5, Android, iOS, Windows, jQuery Mobile, Hadoop, CQ5, Google App Engine, Amazon Web Services, Java, ASP.NET, Python, Rails, Salesforce, Magento etc.

QBurst Technologies is well known for its Global presence as well. QBurst Technologies is a software solutions and consulting company which is spread across the globe which has got branches in USA, UK, Poland, India, Singapore, Australia and UAE.

1. What is QTracker?
   QTracker is an application that helps to track down all the testing devices used in the company with the help of beacons.

2. Why do the employees in QBurst need QTracker?
   The employees use QTracker to track down the testing devices and also to know the precise location and the name of the employee using the particular testing device. Employees can also request the devices and keep track of its progress as and when required. Various features like knowing the number of devices within the vicinity are also incorporated.

3. How does the app keep itself updated on the location?
   The beacons located in different parts of the premises continuously emit Low Bluetooth Energy signals. When the testing device comes under the range of a particular beacon, messages are passed between the beacon and the device and thus the time and other details are stored in the database and updation is made using API calls.

4. How does QTracker keep track of the user who is using the device?
   When the user requires a tsesting device, he should log in using his log in credentials. This change will be updated in the database using API calls.

5. Is QTracker free to use?
   Yes, QTracker application requires only an Internet connection to run, which should be provided by the wireless network in the company.

6. What are the limitations of this application?
   The limitation is that QTracker is restricted to the premises of the company.

I have undergone the training for fourty five days (from 20th June 2016 to 5th August 2016). The training schedule is given below.

Day	Topics Covered
Day 1	Introduction about Environment Setup, Basics of Android
Day 2 - Day 10	Building sample Android application and familiarising with API calls
Day 11 - Day 15	Project requirements and work division
Day 16 - Day 20	Design and Analysis
Day 21 - Day 30	Building individual modules and collaborating work using GitHub
Day 31 - Day 40	Error Handling, Bugs Debugging
Day 41 - Day 45	Testing and Hands-on

Table 1: Training Schedule

3 Life at the Industry

We were divided into groups of 3 and each group was assigned a project each. My project was on Android platform and it was an Android Application based on Estimote Beacons which consisted of modules like Statistics Module, Reminder Module, Profile Module, Settings Module etc. The works were evenly distributed. We had so much fun doing the project and during the process, we could learn a lot about how projects are completed and about the internal workings of the company.

QBurst Technologies is not some place where you get stressed over your work. This is a place which ensures that you feel relaxed while you are on your work. They have got a flat hierarchy structure, where in even a fresher can walk in and talk to the vice president of the company as if they are close buddies. This cool work culture ensures that the productivity of the employee is utilized at its maximum without actually pressing him/her too much.

In fact, their Employees First philosophy is aimed at placing every employee at the forefront of transformation, empowering them with knowledge, resources and infrastructure in a workplace that nurtures innovation. At the same time, they believe that helping employees find work-life continuity, advancing their social and cultural aspirations and providing them an opportunity to connect with each other reduces stress, builds team spirit and optimizes performance.

While I was at QBurst Technologies, working as an intern, I was amazed at the friendships the employees had developed with each other. There was unity in diversity. People from different backgrounds were working together, cracking jokes and enjoying their time together, with a single goal in their minds, to get their project deadlines be met at the right time.

I was provided with proper guidance by a team of resourceful mentors. If I got stuck at my work, I could always ask my teammates for help. If even they

couldnt help, we could always go to our mentors for help. I could study about a lot of new technologies and we have incorporated many of these technologies into our project as well.

3.1 Diversity

QBurst Technologies works hard to create a supportive environment for employees with diverse backgrounds. Accordingly, the company reaches out to a diverse talent pool of different nationalities, cultures, social background and work experiences. Team leaders are equipped with skills to work across the globe in virtual teams. Affinity networks and Employee First Councils help the company reach out to employees from diverse backgrounds. The networks ensure that diverse perspectives are included in all business operations.

3.2 Space - The QBurst Network

QBurst Technologies has its network, named Space, through which updates regarding the company reach the QBurst employees spread across the Globe. Qburst employees connect with each other easily through Space and this contributes a lot to the friendships shared by the QBees

4 Internship Days at QBurst

4.1 Day 1

4.1.1 Orientation Classes

On the first day at QBurst Technologies, we were given an orientation class about the company. A video conference was set up, with which we communicated with QBurst employees from different parts of the world. We were introduced to the mentors and to the rest of the staff. On our first day, we were divided into groups of 3. They were Android, iOS and Back-end.

4.2 Day 2 - Day 10

During the first week, the basic Android concepts were introduced and we were asked to learn by creating sample apps on our own. It was a great learning experience. The topics covered during this period were:

4.2.1 Android Operating System

Android is a mobile operating system (OS) currently developed by Google, based on the Linux kernel and designed primarily for touchscreen mobile devices such as smartphones and tablets. Androids user interface is mainly based on direct manipulation, using touch gestures that loosely correspond to real-world actions, such as swiping, tapping and pinching, to manipulate on-screen objects, along with a virtual keyboard for text input. In addition to touchscreen devices, Google has further developed Android TV for televisions, Android Auto for cars, and Android Wear for wrist watches, each with a specialized user interface. Variants of Android are also used on notebooks, game consoles, digital cameras, and other electronics.

Android applications run in a sandbox, an isolated area of the system that does not have access to the rest of the systems resources, unless access permissions are explicitly granted by the user when the application is installed. Before installing an application, Play Store displays all required permissions: a game may need to enable vibration or save data to an SD card, for example, but should not need to read SMS messages or access the phonebook. After reviewing these permissions, the user can choose to accept or refuse them, installing the application only if they accept. The sandboxing and permissions system lessens the impact of vulnerabilities and bugs in applications, but developer confusion and limited documentation has resulted in applications routinely requesting unnecessary permissions, reducing its effectiveness. Google has now pushed an update to Android Verify Apps feature, which will now run in background to detect malicious processes and crack them down.

4.2.2 Activities and Events

An activity represents a single screen with a user interface just like window or frame of Java. Android activity is the subclass of ContextThemeWrapper class. If you have worked with C, C++ or Java programming language then you must have seen that your program starts from main() function. Very similar way, Android system initiates its program with in an Activity starting with a call on onCreate() callback method. There is a sequence of callback methods that start up an activity and a sequence of callback methods that tear down an activity.

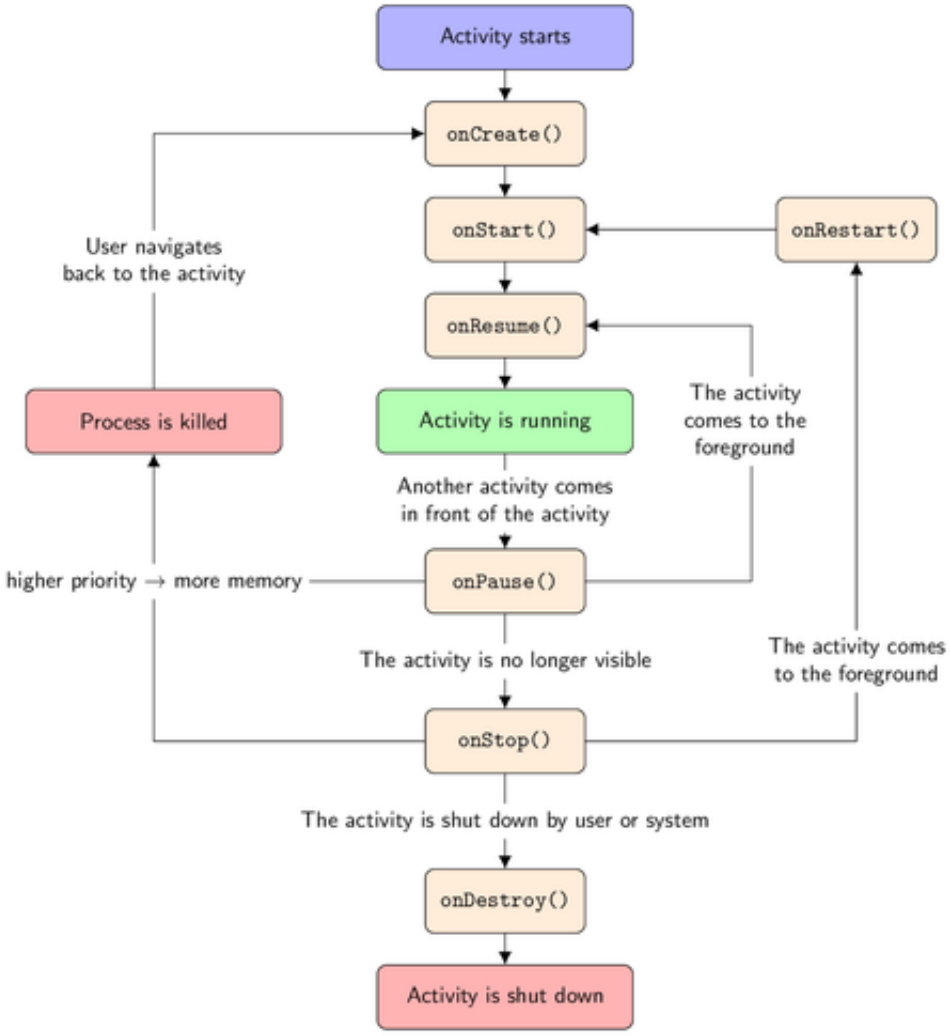


Figure 1: Activity Life-Cycle Diagram

The Activity class defines events or callbacks. You dont need to implement all the callbacks methods. However, its important that you understand each one and implement those that ensure your app behaves the way users expect.

Events are a useful way to collect data about a users interaction with interactive components of Applications. Like button presses or screen touch etc. The Android framework maintains an event queue as first-in, first-out (FIFO)

basis. You can capture these events in your program and take appropriate action as per requirements. An event listener is an interface in the View class that contains a single callback method. These methods will be called by the Android framework when the View to which the listener has been registered is triggered by user interaction with the item in the UI. Event Registration is the process by which an Event Handler gets registered with an Event Listener so that the handler is called when the Event Listener fires the event. Users can interact with their devices by using hardware keys or buttons or touching the screen.Touching the screen puts the device into touch mode. The user can then interact with it by touching the on-screen virtual buttons, images, etc.You can check if the device is in touch mode by calling the View classs isInTouchMode() method. When an event happens and we have registered an event listener for the event, the event listener calls the Event Handlers, which is the method that actually handles the event.

Callback	Description
onCreate()	This is the first callback and called when the activity is first created.
onStart()	This callback is called when the activity becomes visible to the user.
onResume()	This is called when the user starts interacting with the application.
onPause()	The paused activity does not receive user input and cannot execute any code and called when the current activity is being paused and the previous activity is being resumed.
onStop()	This callback is called when the activity is no longer visible.
onDestroy()	This callback is called before the activity is destroyed by the system.
onRestart()	This callback is called when the activity restarts after stopping it.

Figure 2: Callbacks and Descriptions

4.2.3 Services

A service is a component that runs in the background to perform longrunning operations without needing to interact with the user and it works even if application is destroyed. A service can essentially take two states: Started and Bound. A service is started when an application component, such as an activity, starts it by calling startService(). Once started, a service can run in the

background indefinitely, even if the component that started it is destroyed. A service is bound when an application component binds to it by calling bindService(). A bound service offers a client-server interface that allows components to interact with the service, send requests, get results, and even do so across processes with interprocess communication (IPC).

4.2.4 Broadcast Receivers

Broadcast Receivers simply respond to broadcast messages from other applications or from the system itself. These messages are sometime called events or intents. For example, applications can also initiate broadcasts to let other applications know that some data has been downloaded to the device and is available for them to use, so this is broadcast receiver who will intercept this communication and will initiate appropriate action. There are following two important steps to make BroadcastReceiver works for the system broadcasted intents Creating the broadcast receiver and Registering broadcast receiver.

4.2.5 Content Providers

A content provider component supplies data from one application to others on request. Such requests are handled by the methods of the ContentResolver class. A content provider can use different ways to store its data and the data can be stored in a database, in files, or even over a network. Content providers let you centralize content in one place and have many different applications access it as needed.

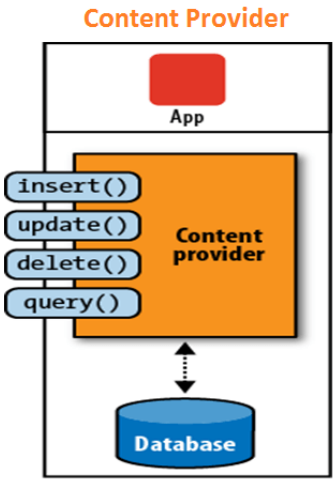


Figure 3: Content Provider

A content provider behaves very much like a database where you can query it, edit its content, as well as add or delete content using insert(), update(),

delete(), and query() methods. In most cases this data is stored in an SQlite database. A content provider is implemented as a subclass of ContentProvider class and must implement a standard set of APIs that enable other applications to perform transactions.

4.2.6 Fragments

A Fragment is a piece of an activity which enable more modular activity design. It will not be wrong if we say, a fragment is a kind of sub-activity. Following are important points about fragment:

- A fragment has its own layout and its own behaviour with its own life cycle callbacks.
- You can add or remove fragments in an activity while the activity is running.
- You can combine multiple fragments in a single activity to build a multiplane UI.
- A fragment can be used in multiple activities.
- Fragment life cycle is closely related to the life cycle of its host activity which means when the activity is paused, all the fragments available in the activity will also be stopped.
- A fragment can implement a behaviour that has no user interface component.
- Fragments were added to the Android API in Honeycomb version of Android which API version 11.

4.2.7 Intents

An Android Intent is an abstract description of an operation to be performed. It can be used with startActivity to launch an Activity, broadcastIntent to send it to any interested BroadcastReceiver components, and startService(Intent) or bindService(Intent, ServiceConnection, int) to communicate with a background Service. The intent itself, an Intent object, is a passive data structure holding an abstract description of an operation to be performed. For example, lets assume that you have an Activity that needs to launch an email client and sends an email using your Android device.

An Intent object is a bundle of information which is used by the component that receives the intent as well as information used by the Android system. Explicit intent going to be connected internal world of application,suppose if you wants to connect one activity to another activity, we can do this quote by explicit intent, below image is connecting first activity to second activity by clicking button. These intents do not name a target and the field for the

component name is left blank. Implicit intents are often used to activate components in other applications.

4.2.8 Testing

The Android framework includes an integrated testing framework that helps you test all aspects of your application and the SDK tools include tools for setting up and running test applications. Whether you are working in Eclipse with ADT or working from the command line, the SDK tools help you set up and run your tests within an emulator or the device you are targeting. Androids build and test tools assume that test projects are organized into a standard structure of tests, test case classes, test packages, and test projects.

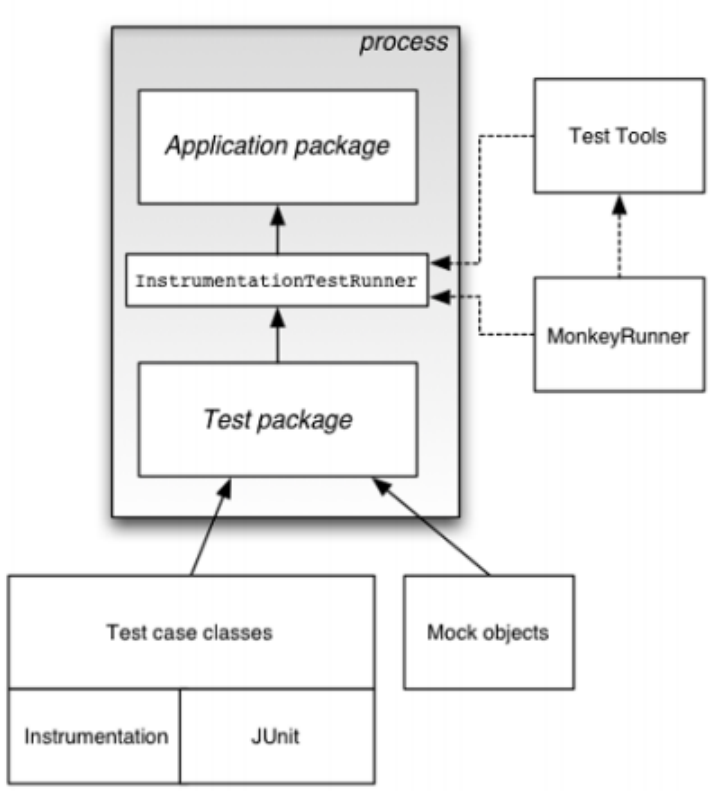


Figure 4: Test Structure

There are many tools that can be used for testing android applications. Some are official like Junit, Monkey and some are third party tools that can be used to test android applications. You can use the JUnit TestCase class to do unit testing on a class that doesnt call Android APIs. TestCase is also the base class for AndroidTestCase, which you can use to test Androiddependent objects. Besides providing the JUnit framework, AndroidTestCase offers Android-specific setup, teardown, and helper methods.

4.3 Day 11- Day 15

We were working on the QTracker project.It is an Android application which tracks all the testing devices in the company. It mainly deals with the new trending technology Internet Of Things(IOT). This application uses Beacon technology. Beacon is a low power energy bluetooth device.

First we discussed with the employees in the company to collect the requirements of the application. We generalized the requirements into features. Following were the features of the application:

- View the list of testing devices
- Register/login to authenticate the employee
- View the list of devices with its details including image, employee name, current device location, android version and battery information
- View the list of beacons along with the devices within its range
- Transfers the ownership of the testing device from one employee to another
- Sign out will be restricted to the box beacon region
- Basic handling of network connectivity issues throughout the application.
- Beacon detection and update the server with the location of the device as the device changes its location.

4.4 Day 16 - Day 30

We created the wireframe for the features. The designer in the QBurst provided the design of the QTracker application according to the features. So we, among the android group selected some features and were working on these features. Before starting all our coding, we learned to use GitHub, a version controller. It allowed and helped us a lot to save our work in remote and could easily integrate all the works.

4.4.1 Login And Registration

Login page is the screen asking your credentials to login to QTracker application. We used two TextView in this page for asking username and password of the user. Define a button with login text and assigned onClick Property to it. After that defined the function mentioned in the onClick property in the java file. The input Type of password TextView was set to password. In the java file, inside the method of onClick gets the username and passwords text using getText() and toString() method and match it with the some basic conditions like password length, userid length etc using equals() strlength() functions. If all that conditions are satisfied we make an api call for login activity by sending user credentials. According to the response from the api call,

on a successful login navigates user to the home page after saving the details in shared preference variables. Otherwise shows invalid login credentials message through snack bar. Login page also contains two other options named as Register and Forgot Password. OnClick actions on register and forgot password options navigates the user to Register and Forgot password page respectively.

Register page includes fields named as UserId, UserName, Password, Confirm Password, EmailID, a Register button and Login option. Functioning of this page is same as Login page.

In Forgot Password page there is only one field , UserId. On a valid UserId

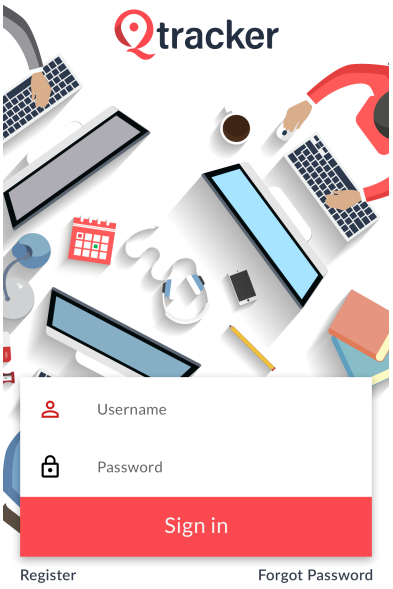


Figure 5: Login and Registration

Forgot button clicks leads user to password reset page. Otherwise shows invalid UserId message through snack bar.

4.4.2 New Device Registration

Unregistered test devices can be registered using this feature. Application can retrieve phone information and status like IMEI number , phone model , battery percent which makes work easier using some standard library function. IMEI can uniquely identify test devices. This feature ensures that only registered devices can use the application.

4.4.3 Navigation Drawer

The navigation drawer is a panel that displays the apps main navigation options on the left edge of the screen. It is hidden most of the time, but is revealed

when the user swipes a finger from the left edge of the screen or, while at the top level of the app, the user touches the app icon in the action bar.

Here the navigation options or menu items are beacons list, devices and

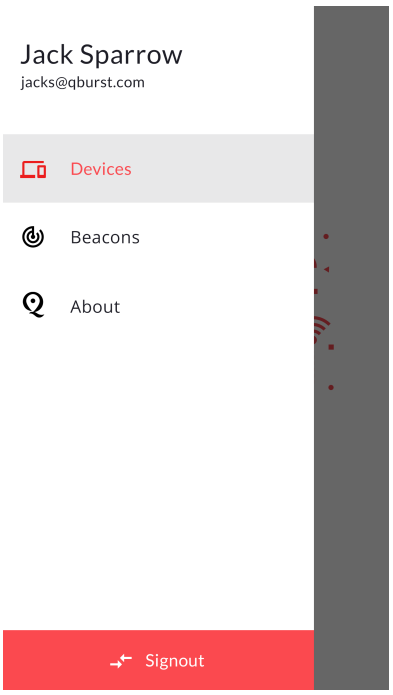


Figure 6: Navigation Drawer

about. Beacon option shows different beacons and devices under it. Devices shows various test devices used and their history of usage. About item gives details about device.

Header of the navigation drawer include the login email and user name. Footer shows sign out option under a particular beacon.Sign out option will be visible only if the device is in the range of Beacon " box " , otherwise the option will be transfer . This option allows the user to Signout from the testing device. On successful Sign out , shows that message through snack bar. Transfer option allows one user to transfer testing device to another user without performing Signout. Login of new user is considered as the signing out function of the old user .

4.4.4 List Of Devices With History

This feature includes the list of testing devices in the company with the user history as well as the location history. The history shows details of the user, location, time and duration.5 of its use. The current user appears at the top in the history. A simple tab layout is used in the user interface for quick and easy access by the user.

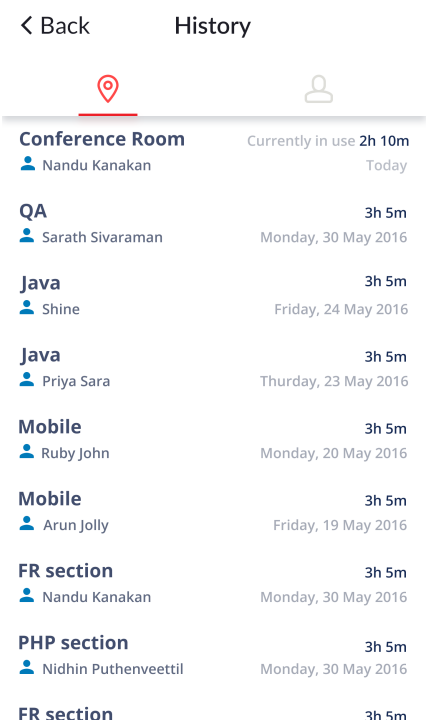


Figure 7: Beacon History

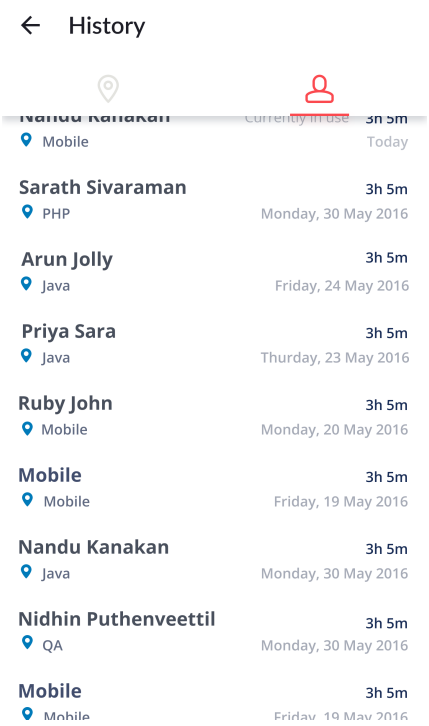


Figure 8: User History

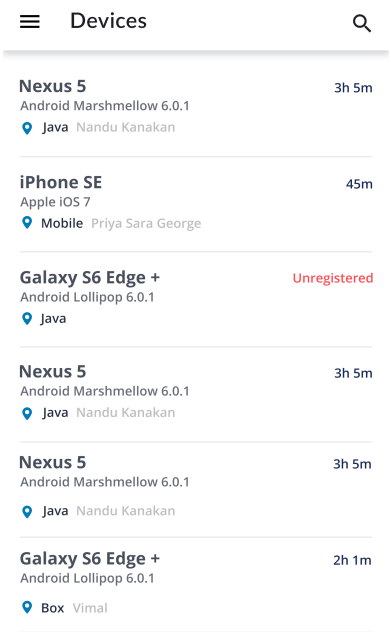


Figure 9: List of Devices

4.4.5 Beacon Detection And Ranging

The device details and the Beacon name are shown in the Home Screen. The bluetooth of the device will be always active. Box, Php, java, QA are the different types of beacon names. A beacon is fixed at every major places in the company. Beacon emits radio waves. So whenever a bluetooth device (that is testing device) moves near the beacon, it would capture the device details and update the history of the user and the beacon by calling APIs.

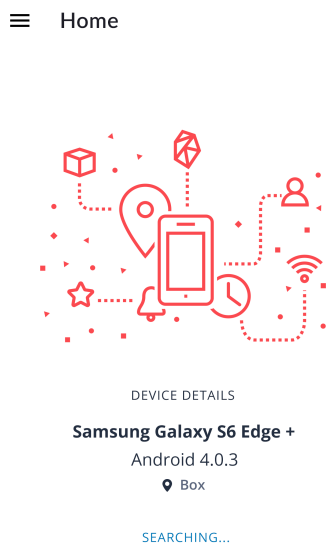


Figure 10: Home Screen

A Beacon is uniquely identified by its UUID, major Id and minor Id. Beacon monitoring can be think as a geofence, i.e., a virtual barrier thats usually defined using a set of geographic coordinates. Moving and out of the area it encloses triggers enter and exit events, which the app can react to. To enable beacon monitoring, first create beaconManager. While monitoring creates a virtual fence to detect when device is moving in and out, ranging actively scans for any nearby beacons and delivers results in every second.

4.4.6 List of Beacons and Devices in each Beacon

This feature includes the list of beacons that are placed in various locations of the company. It also displays the number of devices present within these beacons at that particular time. On clicking a given beacon it enlists the devices that are currently present within the range of that beacon. Moreover, search feature is also implemented so as to enable the user to search for a device within a beacon by entering the device name as the search query.

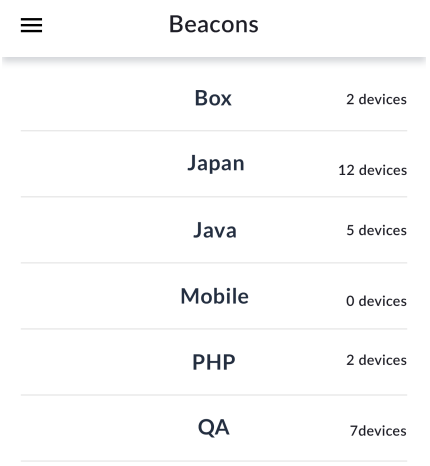


Figure 11: List of Beacons

Figure 12: Devices At the Beacon

4.5 Day 31 - Day 45

The Quality and Assurance (QA) phase was executed thoroughly well. The QA team of QBurst Technologies helped us find the bugs in the application and the bugs were fixed successfully.

QTracker was tested on various testing devices. We have implemented crashlytics in the application. So whenever the app crashes, it would send the code line at which the app crashed. It also sends the device details, so that we could easily re-run the application and resolved the issues.

QA and lint plugins in the Android studio helped a lot to clean our code and could resolve the logical errors.

The final application was presented before the evaluation team of QBurst Technologies with the help of a PowerPoint presentation. They appreciated everyone for making a successful application with all the required features.

5 Development Tools

5.1 Android Studio

Android Studio is the official Integrated Development Environment (IDE) for Android platform development. It was announced on May 16, 2013 at the Google I/O conference. Android Studio is freely available under the Apache

License 2.0. Android Studio was in early access preview stage starting from version 0.1 in May 2013, then entered beta stage starting from version 0.8 which was released in June 2014. The first stable build was released in December 2014, starting from version 1.0. Based on JetBrains IntelliJ IDEA software, Android Studio is designed specifically for Android development. It is available for download on Windows, Mac OS X and Linux, and replaced Eclipse Android Development Tools (ADT) as Googles primary IDE for native Android application development.

New features are expected to be rolled out with each release of Android Studio. The following features are provided in the current stable version:

- Gradle-based build support
- Android-specific re-factoring and quick fixes.
- Lint tools to catch performance, usability, version compatibility and other problems.
- ProGuard integration and app-signing capabilities.
- Template-based wizards to create common Android designs and components.
- A rich layout editor that allows users to drag-and-drop UI components, option to preview layouts on multiple screen configurations
- Support for building Android Wear apps.
- Built-in support for Google Cloud Platform, enabling integration with Google Cloud Messaging and App Engine.

5.2 GitHub

GitHub is a web-based Git or version control repository and Internet hosting service. It offers all of the distributed version control and source code management (SCM) functionality of Git as well as adding its own features. It provides access control and several collaboration features such as bug tracking, feature requests, task management, and wikis for every project.

Projects on GitHub can be accessed and manipulated using the standard Git command-line interface and all of the standard Git commands work with it. GitHub also allows registered and non-registered users to browse public repositories on the site. Multiple desktop clients and Git plugins have also

been created by GitHub and other third parties that integrate with the platform.

6 Third Party Libraries Used

6.1 Estimote SDK

The Estimote SDK for Android is a library that allows interaction with Estimote beacons stickers. The SDK system works on Android 4.3 or above and requires device with Bluetooth Low Energy (SDKs min Android SDK version is 9).

It allows for:

- beacon ranging (scans beacons and optionally filters them by their properties)
- beacon monitoring (monitors regions for those devices that have entered/exited a region)
- nearables (aka stickers) discovery (see quickstart)
- Eddystone scanning (see quickstart)
- easy way to meet all requirements for beacon detection (runtime permissions, acquiring all rights)
- beacon characteristic reading and writing (proximity UUID, major minor values, broadcasting power, advertising interval), see BeaconConnection class and demos in the SDK

6.2 Picasso

Images add much-needed context and visual flair to Android applications. Picasso allows for hassle-free image loading in your applicationoften in one line of code!

Many common pitfalls of image loading on Android are handled automatically by Picasso:

- Handling ImageView recycling and download cancelation in an adapter.
- Complex image transformations with minimal memory use.
- Automatic memory and disk caching.

6.3 Retrofit HTTP Library

Retrofit is a REST Client for Android and Java by Square. It makes it relatively easy to retrieve and upload JSON (or other structured data) via a REST based webservice. In Retrofit you configure which converter is used for the data serialization. Typically for JSON you use GSon, but you can add custom converters to process XML or other protocols. Retrofit uses the OkHttp library for HTTP requests. To work with Retrofit, there are basically three classes:

- Model class which is used to map the JSON data to
- Interfaces which defines the possible HTTP operations
- Retrofit.Builder class - Instance which uses the interface and the Builder API which allows defining the URL end point for the HTTP operation.

Every method of an interface represents one possible API call. It must have a HTTP annotation (GET, POST, etc.) to specify the request type and the relative URL. The return value wraps the response in a Call object with the type of the expected result.

7 Conclusion

The Industrial Training at QBurst Technologies provided an opportunity to apply theoretical knowledge acquired in the classroom with practical application of knowledge required to perform a task. The industrial training taught how the environment will be when we go to work in the real Industry. This training gave an exposure on the tools used in the industry. None of the advanced tools used in the industry were part of the curriculum. This made us understand what our value addition is and what value we can give to the industry.

During the training, I was able to successfully complete an Android project with the co-operation of my teammates Pratheesh M , Musfar P Saleem, Pratheek P, Rahul MR and with the help of my mentors at QBurst Technologies. All the phases of the softwares lifecycle were carefully followed and the result was a successful application. At the end of the training, I felt like I got job oriented. It was such a great experience and I feel like there should be more than one Industrial Training made mandatory in the BTech curriculum.

8 Contact Details

QBurst Technologies
Nisagandhi,
Infopark,
Koratty,
Trissur - 680308.
Website : http://www.qburst.com/
Tel : +91-471-653-7510

Bibliography

[1] "Tutorials Point",
http://www.tutorialspoint.com[Online],Available:http://www.tutorialspoint.com/android

[2] "QBurst Technologies",
https://www.qburst.com/

[3] "GitHub",
https://github.com/