



## Chapter 1

### COMPANY INTRODUCTION

#### 1.1 About company

**Company name:** Runshaw Technologies Pvt. Ltd.

**Registered address:** #142/A 3rd floor, Second main road, Gokulam Second stage, Mysuru-570002

**Official website:** [www.runshaw.in](http://www.runshaw.in)

Telephone number: +91 7892640654

Email ID: [hr@runshaw.in](mailto:hr@runshaw.in)

**Company profile:** Runshaw Technologies Pvt. Ltd, a multi-faceted and fast-growing IT services company in Mysore and Mandya. Established in January 2018 and headquartered in Mysore with other locations in Mandya and Bengaluru. Runshaw Technologies offers high quality and professional IT solutions and services to meet the evolving needs of businesses across the globe. We deliver innovative web solutions and Android development to clients by helping them identify their most important business and technology opportunities and capitalizing on them. We also provide solutions to help them overcome their most complex and critical challenges.

Knowing and understanding a customer's unique needs and being able to fulfil them not just to their satisfaction but also often beyond expectation is the hallmark of a customer centric.Organization.

At Runshaw Technologies, we are committed to achieving and exceeding this level of performance at all times. We are single-mindedly devoted to providing the most reliable and highest-quality of products and services in the realm of IT services – from domain registration and web hosting to web site design and web-based software development, android app development backed by unmatched customer support and service.

In short, we are a one-stop IT Solutions Company. Our web development services aims at executing dynamic applications that would meet the growing business needs on the web. Our technical capabilities allow us to deliver development of every complexity: from simple scripts to complex applications. We have introduced numbers of web sites and web based applications for companies of all sizes. As a digital solutions provider, we

help clients bring the right combination of strategy, experience, design and technology to

every aspect of their web based projects. We offer a wide array of expertise and services to ensure clients achieve innovative and intelligent mobile applications, software and enterprise integration. The smartphone market is evolving very rapidly, at Runshaw, we provide you with expert solutions which help you evolve with this growing market. We develop apps which are cutting edge in technology and are easy to use. We specialize in Android application development and have a dedicated and specialized mobile development and design team. Our services will help you achieve ability to design, develop and deploy successful Android applications which will help you reach millions of Android smartphone users.

## **1.2 We Provide Services in the Following Areas**

- Business & m-commerce (magneto)
- Publishing Application
- Media intensive Application
- Hospitality Application
- Retail Application
- Education Application
- Social Networking Service Application
- Security, Branding, Location based Application
- GPS and Maps integration Application
- Media integration/Music, pictures, videos Application
- Large data management/Streaming and indexing Application

Suitable of Mobile devices and tablets irrespective of make Augmented Reality Applications.

## Chapter 2

### DEPARTMENT DETAILS

#### 2.1 Departments

The department included in the company are

- Testing department
- Development area
- Requirement collection area

##### 2.1.1 Testing department

Software testing is an investigation conducted to provide stakeholders with information about the quality of the software product or service under test. Software testing can also provide an objective, independent view of the software to allow the business to appreciate and understand the risks of software implementation. Test techniques include the process of executing a program or application with the intent of finding software\_bugs (errors or other defects), and verifying that the software product is fit for use.

Software testing involves the execution of a software component or system component to evaluate one or more properties of interest. In general, these properties indicate the extent to which the component or system under test:

- Meets the requirements that guided its design and development.
- Responds correctly to all kinds of inputs.
- Performs its functions within an acceptable time.
- It is sufficiently usable.
- Can be installed and run in its intended environments.
- Achieves the general result its stakeholders desire.

As the number of possible tests for even simple software components is practically infinite, all software testing uses some strategy to select tests that are feasible for the available time and resources. As a result, software testing typically (but not exclusively) attempts to execute a program or application with the intent of finding software\_bugs (errors or other defects). The job of testing is an iterative process as when one bug is fixed, it can illuminate other, deeper bugs, or can even create new ones.

## TESTING APPROACHES

- **Static vs dynamic testing:** There are many approaches available in software testing. Reviews, walkthroughs, or inspections are referred to as static testing, whereas executing programmed code with a given set of test cases is referred to as dynamic testing. Static testing is often implicit, like proofreading, plus when programming tools/text editors check source code structure or compilers (pre-compilers) check syntax and data flow as static program analysis. Dynamic testing takes place when the program itself is run. Dynamic testing may begin before the program is 100% complete in order to test particular sections of code and are applied to discrete functions or modules. Typical techniques for these are either using stubs/drivers or execution from a debugger environment. Static testing involves verification, whereas dynamic testing also involves validation.
- **Exploratory approaches:** Exploratory testing is an approach to software testing that is concisely described as simultaneous learning, test design and test execution. Cem Kaner, who coined the term in 1984, defines exploratory testing as "a style of software testing that emphasizes the personal freedom and responsibility of the individual tester to continually optimize the quality of his/her work by treating test-related learning, test design, test execution, and test result interpretation as mutually supportive activities that run in parallel throughout the project.
- **The box approach:** Software testing methods are traditionally divided into white- and black-box testing. These two approaches are used to describe the point of view that the tester takes when designing test cases. A hybrid approach called grey-box testing may also be applied to software testing methodology. With the concept of grey-box testing which develops tests from specific design elements gaining prominence, this "arbitrary distinction" between black- and white-box testing has faded somewhat.
- **White-box testing,** white-box testing verifies the internal structures or workings program, as opposed to the functionality exposed to the end-user. In white-box testing, an internal perspective of the system (the source code), as well as programming skills, are used to design test cases. The tester chooses inputs to exercise paths through the code and determine the appropriate outputs. This is analogous to testing nodes in a circuit, e.g., incircuit testing (ICT).

- **Black-box testing**, black-box testing (also known as functional testing) treats the software as a "black box," examining functionality without any knowledge of internal implementation, without seeing the source code. The testers are only aware of what the software is supposed to do, not how it does it. Black-box testing methods include: equivalence partitioning, boundary value analysis, all-pairs testing, state transition tables, decision table testing, fuzz testing, model-based testing, use case testing, exploratory testing, and specification-based testing.
- **Component interface testing**, component interface testing is a variation of black-box testing, with the focus on the data values beyond just the related actions of a subsystem component. The practice of component interface testing can be used to check the handling of data passed between various units, or subsystem components, beyond full integration testing between those units. The data being passed can be considered as "message packets" and the range or data types can be checked, for data generated from one unit, and tested for validity before being passed into another unit. One option for interface testing is to keep a separate log file of data items being passed, often with a timestamp logged to allow analysis of thousands of cases of data passed between units for days or weeks. Tests can include checking the handling of some extreme data values while other interface variables are passed as normal values. Unusual data values in an interface can help explain unexpected performance in the next unit.
- **Visual testing**, the aim of visual testing is to provide developers with the ability to examine what was happening at the point of software failure by presenting the data in such a way that the developer can easily find the information she or he requires, and the information is expressed clearly.
- **Grey-box testing**, grey-box testing (American spelling: gray-box testing) involves having knowledge of internal data structures and algorithms for purposes of designing tests while executing those tests at the user, or black-box level. The tester will often have access to both "the source code and the executable binary." Grey-box testing may also include reverse engineering (using dynamic code analysis) to determine, for instance, boundary values or error messages. Manipulating input data and formatting output do not qualify as grey-box, as the input and output are clearly outside of the "black box" that we are calling the system under test. This distinction is particularly important when

conducting integration testing between two modules of code written by two different developers, where only the interfaces are exposed for the test.

## TESTING LEVELS

There are at least three levels of testing: unit testing, integration testing, and system testing. However, a fourth level, acceptance testing, may be included by developers. This may be in the form of operational acceptance testing or be simple end-user (beta) testing, testing to ensure the software meets functional expectations. Tests are frequently grouped into one of these levels by where they are added in the software development process, or by the level of specificity of the test.

- **Unit testing:** Unit testing refers to tests that verify the functionality of a specific section of code, usually at the function level. In an object-oriented environment, this is usually at the class level, and the minimal unit tests include the constructors and destructors. These types of tests are usually written by developers as they work on code (white-box style), to ensure that the specific function is working as expected. One function might have multiple tests, to catch corner cases or other branches in the code. Unit testing alone cannot verify the functionality of a piece of software, but rather is used to ensure that the building blocks of the software work independently from each other.

Unit testing is a software development process that involves a synchronized application of a broad spectrum of defect prevention and detection strategies in order to reduce software development risks, time, and costs. It is performed by the software developer or engineer during the construction phase of the software development life cycle. Unit testing aims to eliminate construction errors before code is promoted to additional testing; this strategy is intended to increase the quality of the resulting software as well as the efficiency of the overall development process. Depending on the organization's expectations for software development, unit testing might include static code analysis, data-flow analysis, metrics analysis, peer code reviews, code coverage analysis and other software testing practices.

- **Integration testing:** Integration testing is any type of software testing that seeks to verify the interfaces between components against a software design. Software components may



be integrated in an iterative way or all together ("big bang"). Normally the former is considered a better practice since it allows interface issues to be located more quickly and fixed.

Integration testing works to expose defects in the interfaces and interaction between integrated components (modules). Progressively larger groups of tested software components corresponding to elements of the architectural design are integrated and tested until the software works as a system.

- **System testing:** System testing tests a completely integrated system to verify that the system meets its requirements. For example, a system test might involve testing a logon interface, then creating and editing an entry, plus sending or printing results, followed by summary processing or deletion (or archiving) of entries, then logoff.

### 2.1.2 Development Area

Software development is the process of conceiving, specifying, designing, programming, documenting, testing, and bug fixing involved in creating and maintaining applications, frameworks, or other software components. Software development is a process of writing and maintaining the source code, but in a broader sense, it includes all that is involved between the conception of the desired software through to the final manifestation of the software, sometimes in a planned and structured process. Therefore, software development may include research, new development, prototyping, modification, reuse, re-engineering, maintenance, or any other activities that result in software products.

Software can be developed for a variety of purposes, the three most common being to meet specific needs of a specific client/business (the case with custom software), to meet a perceived need of some set of potential users (the case with commercial and open source software), or for personal use. Embedded software development, that is, the development of embedded software, such as used for controlling consumer products, requires the development process to be integrated with the development of the controlled physical product. System software underlies applications and the programming process itself, and is often developed separately.

The need for better quality control of the software development process has given rise to the discipline of software engineering, which aims to apply the systematic approach exemplified

in the engineering paradigm to the process of software development. There are many approaches to software project management, known as software development life cycle models, methodologies, processes, or models. The waterfall model is a traditional version, contrasted with the more recent innovation of agile software development.

Most methodologies share some combination of the following stages of software development:

- Analyzing the problem
- Market research
- Gathering requirements for the proposed business solution
- Devising a plan or design for the software-based solution
- Implementation (coding) of the software
- Testing the software
- Deployment
- Maintenance and bug fixing

## SOFTWARE DEVELOPMENT ACTIVITIES

- **Identification of need:** The sources of ideas for software products are plentiful. These ideas can come from market research including the demographics of potential new customers, existing customers, sales prospects who rejected the product, other internal software development staff, or a creative third party. Ideas for software products are usually first evaluated by marketing personnel for economic feasibility, for fit with existing channels distribution, for possible effects on existing product lines, required features, and for fit with the company's marketing objectives. In a marketing evaluation phase, the cost and time assumptions become evaluated. A decision is reached early in the first phase as to whether, based on the more detailed information generated by the marketing and development staff, the project should be pursued further.
- **Planning:** Planning is an objective of each and every activity, where we want to discover things that belong to the project. An important task in creating a software program is extracting the requirements or requirements analysis. Customers typically have an abstract idea of what they want as an end result but do not know what software should do. Skilled and experienced software engineers recognize incomplete, ambiguous, or even

contradictory requirements at this point. Frequently demonstrating live code may help reduce the risk that the requirements are incorrect.

"Although much effort is put in the requirements phase to ensure that requirements are complete and consistent, rarely that is the case; leaving the software design phase as the most influential one when it comes to minimizing the effects of new or changing requirements. Requirements volatility is challenging because they impact future or already going development efforts." Once the general requirements are gathered from the client, an analysis of the scope of the development should be determined and clearly stated. This is often called a scope document

- **Designing:** Once the requirements are established, the design of the software can be established in a software design document. This involves a preliminary or high-level design of the main modules with an overall picture (such as a block diagram) of how the parts fit together. The language, operating system, and hardware components should all be known at this time. Then a detailed or low-level design is created, perhaps with prototyping as proof-of-concept or to firm up requirements.
- **Implementation, testing and documentation:** Implementation is the part of the process where software engineers actually program the code for the project. Software testing is an integral and important phase of the software development process. This part of the process ensures that defects are recognized as soon as possible. In some processes, generally known as test-driven development, tests may be developed just before implementation and serve as a guide for the implementation's correctness.

Documenting the internal design of software for the purpose of future maintenance and enhancement is done throughout development. This may also include the writing of an API, be it external or internal. The software engineering process chosen by the developing team will determine how much internal documentation (if any) is necessary. Plan-driven models (e.g., Waterfall) generally produce more documentation than Agile models.

- **Deployment and maintenance:** Deployment starts directly after the code is appropriately tested, approved for release, and sold or otherwise distributed into a production environment. This may involve installation, customization (such as by setting

parameters to the customer's values), testing, and possibly an extended period of evaluation. Software training and support is important, as software is only effective if it is used correctly.

Maintaining and enhancing software to cope with newly discovered faults or requirements can take substantial time and effort, as missed requirements may force redesign of the software.

### **2.1.3 Requirement collection area**

When it comes to any type of project, requirement collection plays a key role. Requirements collection is not only important for the project, but it is also important for the project management function. For the project, understanding what the project will eventually deliver is critical for its success. Through requirements, the project management can determine the end deliveries of the project and how the end deliveries should address client's specific requirements.

Although requirements collection looks quite straightforward, surprisingly, this is one of the project phases where most of the projects start with the wrong foot. In general, majority of the failed projects have failed due to the wrong or insufficient requirements gathering.

Some parameters for successful requirements collection process:

- Never assume that you know customer's requirements. What you usually think could be quite different to what the customer wants. Therefore, always verify with the customer when you have an assumption or a doubt.
- Get the end-users involved from the start. Get their support for what you do.
- At the initial levels, define the scope and get customer's agreement. This helps you to successfully focus on scope of features.
- When you are in the process of collecting the requirements, make sure that the requirements are realistic, specific and measurable.
- Focus on making the requirements document crystal clear. Requirement document is the only way to get the client and the service provider to an agreement. Therefore, there should not be any gray area in this document. If there are gray areas, consider this would lead to potential business issues.

- Do not talk about the solution or the technology to the client until all the requirements are gathered. You are not in a position to promise or indicate anything to the client until you are clear about the requirements.
- Before moving into any other project phases, get the requirements document signed off by the client.
- If necessary, create a prototype to visually illustrate the requirements.

Requirement collection is the most important step of a project. If the project team fails to capture all the necessary requirements for a solution, the project will be running with a risk. This may lead to many disputes and disagreements in the future, and as a result, the business relationship can be severely damaged. Therefore, take requirement collection as a key responsibility of the project team. Until the requirements are signed off, do not promise or comment on the nature of the solution.

## **Chapter 3**

### **SYSTEM ARCHITECTURE & IMPLEMENTATION ANALYSIS**

#### **3.1 Existing system and proposed system**

##### **3.1.1 Existing system**

In an existing system, admin will handle entire application like permitting new member to join a group and other groups , posting events schedule..etc.

##### **3.1.2 Disadvantages of existing system**

In an existing system, handling resources is difficult creating a room for mis-communication and mis-understanding since admin is the sole person.

##### **3.1.3 Proposed system**

To design and implement the CLUB MEMBERSHIP MANAGEMENT SYSTEM, which contains all club related records where students/members of a club can register only once and join multiple clubs where their data is handled by admin and moderator , moderator requires permission by the admin to handle related club assigned also can post events.

##### **3.1.4 Advantages of proposed system**

In single application a user can get the necessary information related to clubs , events , members and a good interface to handle new memberships.

## 3.2 Hardware and software requirements

### Software Requirements

- **Language:** HTML, Java Script, CSS, Angular JS.
- **UI Technologies:** HTML5
- **Platform:** Windows 7/8.1/10.
- **Framework:** Visual Studio

### Hardware Requirements

- Hard disk : 40 GB
- RAM : 2 GB
- Processor Speed : 3.00GHz
- Processor : Core i3

## **Chapter 4**

### **REFLECTION NOTES**

#### **4.1 Work experience**

The purposes of the internship is to prepare the student to serve the needs of industry and commerce more effectively upon graduation. This included providing real life experience and exposure, thus gaining first-hand exposure of working in the real world, granting the opportunity to learn more about the intern self-potentials and abilities, getting connected and developing professional network, preventing CV from going to trash during job application and transition to full-time job position.

RunShaw Technology Pvt. Ltd has wide experience both in development platforms and business domains. It does not specialize in particular technologies, but it has definitely proficient and experienced with a wide array of tools and systems. Every product is unique and it tries to fit the right team with the right skills for that particular product. Its teams have worked on building numerous mobile apps.

#### **4.2 Problems/Challenges**

Internship is the process of on-the-job training, which particularly beneficial for students with major in technical courses. Internship allows students to harness the skill, knowledge and theoretical practice they learnt in the college and enabling them to transform theory into practical real life situations. The intern's time with Solutions Technologies Limited has really illuminated his knowledge and understanding of practical experience of the theories being taught in the corporate environment.

In this project I have been selected by Runshaw Technology Pvt Ltd. In the Team of Runshaw is a senior software engineer and we are learned many things from senior developer. From designing software architecture to using software tools effectively, senior developer has taught us everything. I consider myself very lucky to get such a mentor and supervisor. As this is a huge project, most of my team members have been working on it.

The main challenge for me was that I just joined as an intern in Runshaw and had little idea about industry projects. And this project was almost a complete project which I needed to



understand in a short time all by myself. As I was assigned to fix bugs, I needed to have clear knowledge of the operations inside so that I could figure out the wrong logic there and correct it.

It was a real challenge for students.

In today's world, every organization depends heavily on technology to keep business operations connected, drive revenue and support ever growing customer demands. Undoubtedly, Information Technology (IT) is the key to manage technology and it spans wide variety of areas that include but are not limited to things such as computer software, information systems, programming languages, database management systems, networking etc.

### 4.3 Technical Outcomes

As stated before, I had been assigned works in windows platform. Therefore most of the things I learned are Microsoft and its other technologies.

#### Tool

Programming tools make development easier. In my intern at Runshaw Software I have used the following tools in my daily works.

- JAVA Script
- HTML5

#### Technology

##### JAVA Script

JavaScript (often shortened to JS) is a **lightweight, interpreted, object-oriented language with first-class functions**, and is best known as the scripting language for Web pages, but it's used in many non-browser environments as well.

- **Simple:** The java language is simple to learn. Java code is easy to read and write.
- **Familiar:** Java is similar to c/c ++ but it removes the drawbacks and complexities of c/c ++ like pointers and multiple inheritances. So if you have background in c/c ++, you will find java familiar and easy to learn.
- **Object oriented:** Unlike c ++ which is semi object oriented, java is fully object oriented programming language. It has all OOP features such as abstraction, encapsulation, inheritance and polymorphism.

- **Robust:** With automatic garbage collection and simple memory management model, plus language features like generics, try with resources..., java guides programmer toward reliable programming habits for creating highly reliable applications.
- **Secure:** The java platform is designed with security features built into the language and runtime system such as static type-checking at compile time and runtime checking , which let you creating applications that can't be invaded from outside. You never hear about viruses attacking java applications.
- **High performance:** Java code is compiled into bytecode which is highly optimized by the java compiler, so that the java virtual machine can execute java applications at full speed. In addition, compute-intensive code can be re-written in native code and interfaced with java platform via java native interface thus improve the performance.

## HTML5

- HTML5 is a markup language used for structuring and presenting
- HTML5 is important because now that all major vendors support it including those in the mobile space

You get a universal experience across a wide spectrum of devices. While particular features of HTML5 might vary by device platform, there are enough common ground features to build a compelling web app that looks and feels similar across the gamut of devices.

## Framework Visual studio

The Visual Studio team's focus on making the fundamental features and flows of the Integrated Development Environment (IDE) rock-solid, along with refining and polishing the user-facing features that matter to you in your day-to-day app development workflows. In Project Marble, we are specifically looking at reducing the number of crashes, hangs, memory leaks, and user-impacting bugs. We are also investing in our measurement infrastructure to prevent these issues from occurring. Stay tuned for more updates and details as we progress on this initiative.

## 4.4 Non-Technical Out Comes

### Professional Learning

Although technical learning is important, professional learning is the sole purpose of an internship. Runshaw is an excellent place to learn professionalism.

- **No bullying and blaming:** Application development is always a team work. And when there is a team work, misunderstanding is very usual. This is true for Runshaw too. However, we have never seen team leaders and project managers to bully people working under their supervision. Personally, we have made a lot of mistakes last in six months. But my project manager had never been harsh with us. This practice is really effective to keep the work environment healthy. Blaming others for their mistakes does not solve the problem. It only makes the situation and the relationship between coworkers worse.
- **Do your work yourself:** at Runshaw, there is no one to take orders like, “do this work”, and “do that work”. Everyone is assigned to a particular work and he does his work in his way. At time of scrum everyone shares their progress to others. All the projects are done in this way.
- **Appreciate success, do not discourage for failure:** In Runshaw appreciation for success is a real trend. The team I have been assigned to has taught us the value of appreciation. Here, the members appreciate each other on their successful contribution to the company and also on their success in some other fields.
- **Attitude:** Runshaw is a Software Company with full of fun and creative and very much friendly. As an intern these attract us very much and I always try to follow them to be a successful Software Engineer as well as a successful man.
- **Quality of work:** Runshaw follows a great standard of pure software engineering and their product quality is very high. Time to time code is reviewed so that better quality software is developed. We tried to maintain the standard of work from my side.
- **Planning:** Before starting a task I had always planned that how should I approach towards it. Though those plans only included me, at least I can say that I am habituated with planning.
- **Organizing:** One of the best ways of learning how to organize is to start organizing oneself of his/her own and after spending almost in a months at snipe we should say that

we are much more organized person only by practicing that principle. And now being organized our self, we can say that we are ready to organize others.

- **Attendance and Punctuality:** Punctuality is another important issue in professional life. I have to go office at 10:30 to attend scrum. But at every day we have to be there before 11 for attending Team meeting. This practice helped me to increase my level of punctuality and I was present in approximately 99% workdays. **Skills**
- Can acquire and evaluate information - can identify need for data, can obtain it from existing sources or can create it, and can evaluate its relevance and accuracy
- Can organize and maintain information - can organize, process and maintain written or Computerized records and other forms of information in a systematic fashion
- Can interpret and communicate information - can select and analyze information and communicate the results to others using oral, written, graphic, pictorial, or multimedia methods
- Can use computers to process information - can employ computers to acquire, organize, analyze and communicate information
- Can exercise leadership - can communicate thoughts, feelings and ideas to justify a position; can encourage, persuade, convince, or otherwise motivate an individual or group, including responsibly challenging existing procedures, policies or authority
- Can negotiate - can work towards an agreement that may involve exchanging specific resources or resolving divergent interests
- Can work with cultural diversity - can work well with men and women and with a variety of ethnic, social or educational backgrounds.

## 4.5 Benefits of doing internship

- **Gain Valuable Work Experience**

An internship provides the opportunity to gain hands on work experience that is not possible to get in the classroom and also companies train interns and help in gaining the experience required to get a job.

- **Have an Edge in the Job Market**

Employers are usually more concerned with the work experience than the qualifications and internships are often the only way to get the work experience that is required to secure a job, so pursuing internship have a vital role in the resume. Many employers prefer or require Applicants pursuing internship or relevant work experience.

Hence by having an internship experience the resume gets weight age.

- **Transition into a Job**

Employers see interns as prospective employees, so by performing well one can finish internships and continue working with the company full time. Internships are the number one way for employers to find new staffs and employees to find a new job with experience.

- **Networking Opportunities**

Internships are a great way to meet people in specific field of our interest. An internship allows meeting people who might help in getting a job later on and give the contacts of the industry to which we are interested and break into it. Plus, references from people in the industry will really add weight to the application.

- **Apply Classroom Knowledge**

An internship can be seen as the pinnacle of the education and give a chance to use the skills learned in the classroom in a real-world setting. It's a chance to prove the worth of the qualifications and to show the ability to perform a role that has been assigned.

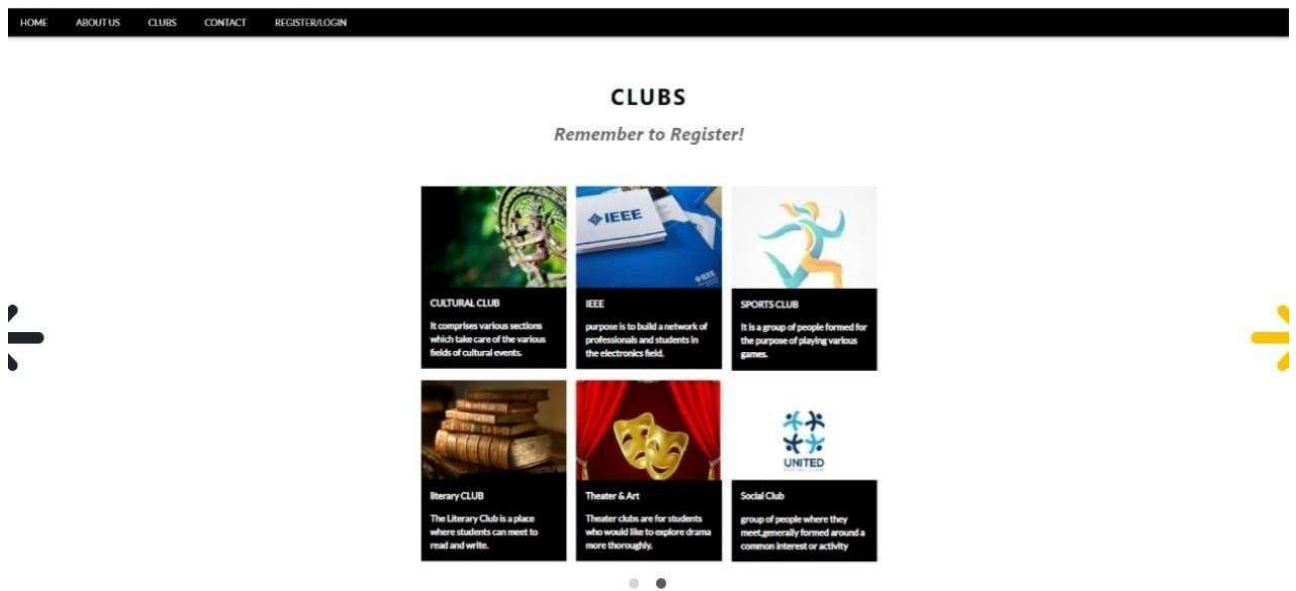
## 4.6 Website Developed during Internship

### 1) Snapshot-1:



HOME page

## 2) Snapshot-2



Club Register Page

## **CONCLUSION**

After completing the internship, I have been exposed to a working life. Along the training period, I realized that observation is a main element to find out the root cause of a problem. Not only for the project but daily activities too. During the project, cooperation is the key element to determine the problems. It taught me to work together as a team and how to contribute to the team with individual efforts. Moreover, the project indirectly helps to learn independently, discipline our self, be considerate/patient, self-trust, take initiative and the ability to solve problems. In sum, the activities that we have learned during internship are useful for us in future to face challenges in a working environment. It also increased my confidence level.



