INTERNATIONAL INSTITUTE OF INFORMATION TECHNOLOGY

SOFTWARE PRODUCTION ENGINEERING PROJECT

CS 816 - SOFTWARE PRODUCTION ENGINEERING

Spandan App

Bishal Pandia
IMT2017010

Tanishq Jaswani
IMT2018077

Phani Tirthala
IMT2017031

May 25, 2021

Contents

1 Abstract

This report gives a brief description on the process of building the Spandan app by Phani Tirthala (IMT2017031), Bishal Pandia (IMT2017010) and Tanishq Jaswani (IMT2018077). The spandan app is nothing but a tournament tracker app that allows users to access any of the on-going tournaments and matches that are happening. It allows the users to be notified about a match that they are interested in. This report also explains how DevOps toolchain was used to deploy the application.

Devops was used extensively to develop features, test, build and deploy the application. We have used flutter for the front end of the app, " " for communicating with the database and "" for the database. We have used Git for version control and Jenkin for Continuous Integration. For continuous deployment and delivery we have used Docker and Ansible.

2 Introduction

2.1 Problem Statement

Let us assume that you were in a sports tournament where you or your friends were participating in. Considering that the tournament would consist of more than one sport's games happening at once, you would like to know when you a particular team's game is going to happen or be notified on when the game would begin. If there's a particular game that you were not able to attend but would like to see the score rather than calling some friend, then wouldn't it be better if you could access that all through the help of just of one app.

Presenting to you the Spandan app, this app allows users to view what all matches are happening in a particular tournament and also the scores of on-going or past matches. This app also notifies users on a game that they would interested in as well.

2.2 Features

The features include

- User authorization that allows players of the teams that are interested in the tournament to be notified about anything regarding their game in hindsight.

- Live score updates of any game for any user to access through the app.
- Notifications to the players as well as users regarding any match they are interested in, any updates for the tournament.

2.3 System Configuration

- GNOME : 3.28.2
- OS Type : 64-bit

2.3.1 Operating System

- Ubuntu 18.04.4 LTS

2.3.2 CPU and RAM

- RAM : 7.7 GiB
- CPU : Intel® Core™ i5-8250U CPU @ 1.60GHz × 8

2.3.3 Languages and Tools

- Flutter
- PHP
- MySQL

2.3.4 DevOps Tools

- Git: Source Control
- Build: Flutter/Gradle
- Jenkins: Continuous Integration
- Docker: Containerization
- Continuous Deployment: Ansible
- Continuous Monitoring: ELK Stack

3 Application

3.1 User Stories

- As a user, I would like to have access to the scores of the current matches.
- As a user, I would like to know the matches that are going on in all the sports of a particular tournament.
- As a user, I would like to know the score of the games that have already gotten over.
- As a user, I would like to be able to sign up into the application to view the results of the tournament.
- As a user, I would like to be able to be able to sign with my credentials being saved.
- As a user, I would like to be able to change my password in case I have forgotten my previous one.

3.2 Use Case Diagram

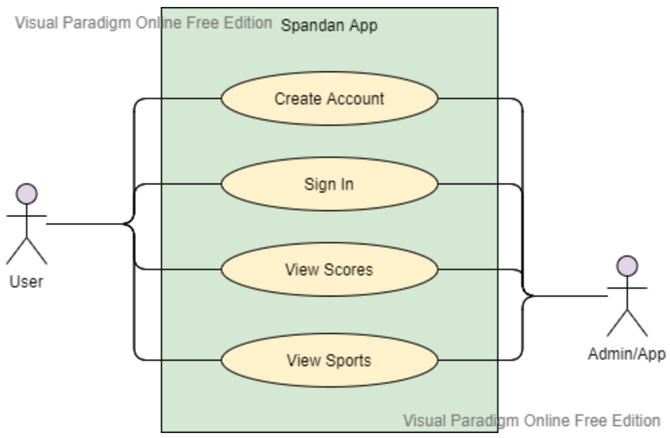

Visual Paradigm Online Free Edition
Spandan App
Create Account
Sign In
View Scores
View Sports
User
Admin/App
Visual Paradigm Online Free Edition

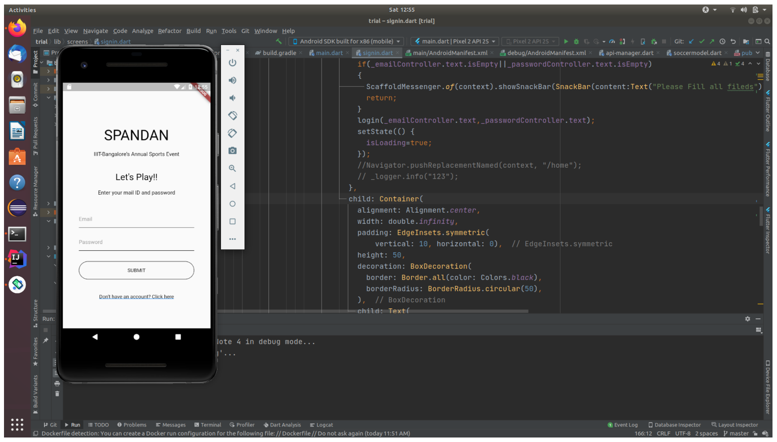
Figure 1: User Stories

3.3 UI

The following are the different screen you encounter while using the application:

3.3.1 Login Screen

- Here we can see the initial login page of the application that contains a text box for email id and password, followed by a link that directs them to a sign up page in case they do not have an account. [2]



```
if(_emailController.text.isEmpty||_passwordController.text.isEmpty)
{
  ScaffoldMessenger.of(context).showSnackBar(SnackBar(content:Text("Please Fill all fileds")
  return;
}
login(_emailController.text,_passwordController.text);
setState(() {
  isLoading=true;
});
//Navigator.pushReplacementNamed(context, "/home");
// _logger.info("123");
},
child: Container(
  alignment: Alignment.center,
  width: double.infinity,
  padding: EdgeInsets.symmetric(
      vertical: 10, horizontal: 0),  // EdgeInsets.symmetric
  height: 50,
  decoration: BoxDecoration(
    border: Border.all(color: Colors.black),
    borderRadius: BorderRadius.circular(50),
  ),  // BoxDecoration
  child: Text(
```

SPANDAN

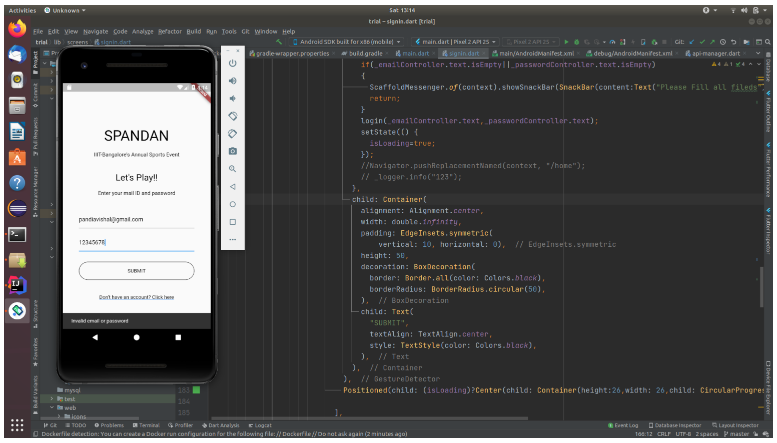
IIIT-Bangalore's Annual Sports Event

Let's Play!!

Enter your mail ID and password

Figure 2: Login Screen

- After entering the details the login screen would would throw an error in case the user login with the incorrect credentials. [3]


SPANDAN

IIIT-Bangalore's Annual Sports Event

Let's Play!!

Enter your mail ID and password

pandiavishal@gmail.com

12345678

SUBMIT

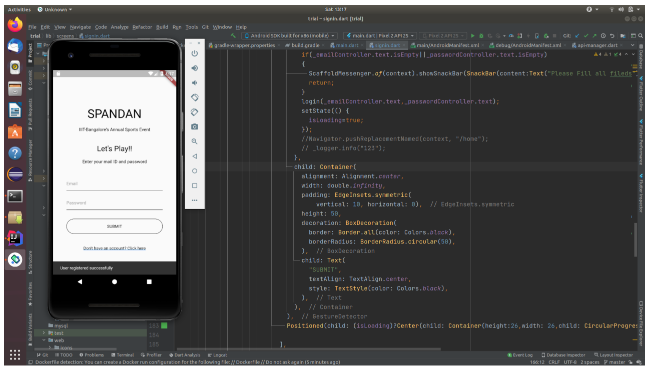
Don't have an account? Click here

Invalid email or password

```
if(_emailController.text.isEmpty||_passwordController.text.isEmpty)
{
  ScaffoldMessenger.of(context).showSnackBar(SnackBar(content:Text("Please Fill all fileds"
  return;
}
login(_emailController.text,_passwordController.text);
setState(() {
  isLoading=true;
});
//Navigator.pushReplacementNamed(context, "/home");
// _logger.info("123");
},
child: Container(
  alignment: Alignment.center,
  width: double.infinity,
  padding: EdgeInsets.symmetric(
      vertical: 10, horizontal: 0),  // EdgeInsets.symmetric
  height: 50,
  decoration: BoxDecoration(
    border: Border.all(color: Colors.black),
    borderRadius: BorderRadius.circular(50),
  ),  // BoxDecoration
  child: Text(
    "SUBMIT",
    textAlign: TextAlign.center,
    style: TextStyle(color: Colors.black),
  ),  // Text
),  // Container
),  // GestureDetector
Positioned(child: (isLoading)?Center(child: Container(height:26,width: 26,child: CircularProgres

],
```

Figure 3: Invalid Login

- Once a user finishes signing up, they will be redirected to the login screen with a message saying that their registration was successful. [4]


SPANDAN

IIIT-Bangalore's Annual Sports Event

Let's Play!!

Enter your mail ID and password

Email

Password

SUBMIT

Don't have an account? Click here

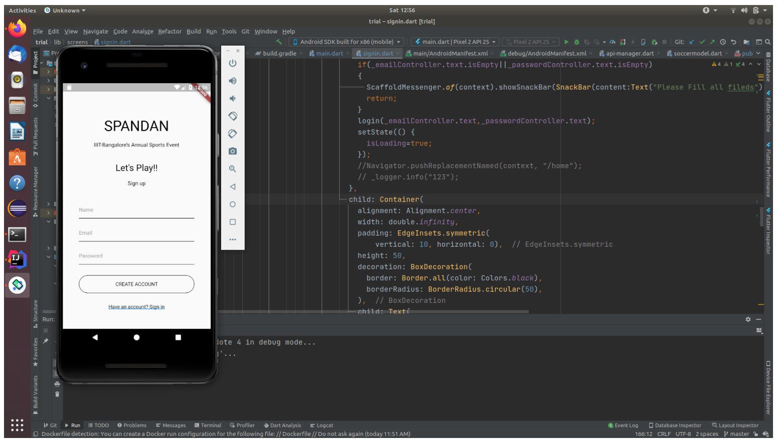
User registered successfully

```
if(_emailController.text.isEmpty||_passwordController.text.isEmpty)
{
  ScaffoldMessenger.of(context).showSnackBar(SnackBar(content:Text("Please Fill all fileds"
  return;
}
login(_emailController.text,_passwordController.text);
setState(() {
  isLoading=true;
});
//Navigator.pushReplacementNamed(context, "/home");
// _logger.info("123");
},
child: Container(
  alignment: Alignment.center,
  width: double.infinity,
  padding: EdgeInsets.symmetric(
      vertical: 10, horizontal: 0),  // EdgeInsets.symmetric
  height: 50,
  decoration: BoxDecoration(
    border: Border.all(color: Colors.black),
    borderRadius: BorderRadius.circular(50),
  ),  // BoxDecoration
  child: Text(
    "SUBMIT",
    textAlign: TextAlign.center,
    style: TextStyle(color: Colors.black),
  ),  // Text
),  // Container
),  // GestureDetector
Positioned(child: (isLoading)?Center(child: Container(height:26,width: 26,child: CircularProgres

],
```

Figure 4: After Registering

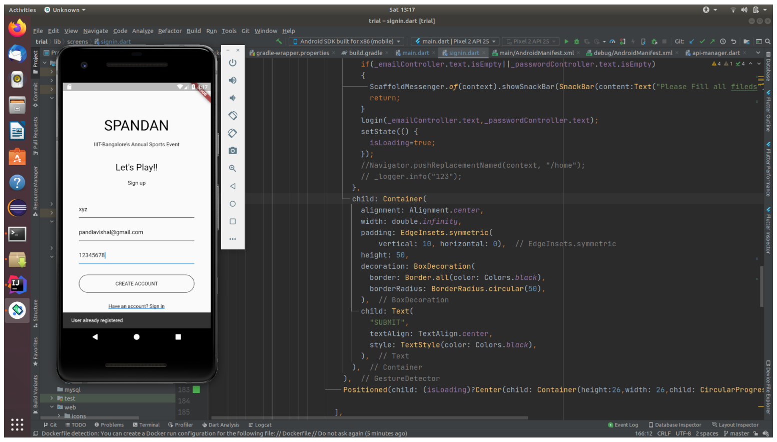
3.3.2 Sign Up Screen

- In the sign up screen the user has to enter their details for which their account would be created and the credentials to be stored. [5] [6]



```
    if(_emailController.text.isEmpty||_passwordController.text.isEmpty)
    {
      ScaffoldMessenger.of(context).showSnackBar(SnackBar(content:Text("Please Fill all fileds")
      return;
    }
    login(_emailController.text,_passwordController.text);
    setState(() {
      isLoading=true;
    });
    //Navigator.pushReplacementNamed(context, "/home");
    // _logger.info("123");
  },
  child: Container(
    alignment: Alignment.center,
    width: double.infinity,
    padding: EdgeInsets.symmetric(
        vertical: 10, horizontal: 0),  // EdgeInsets.symmetric
    height: 50,
    decoration: BoxDecoration(
      border: Border.all(color: Colors.black),
      borderRadius: BorderRadius.circular(50),
    ),  // BoxDecoration
```

Figure 5: Sign Up Screen



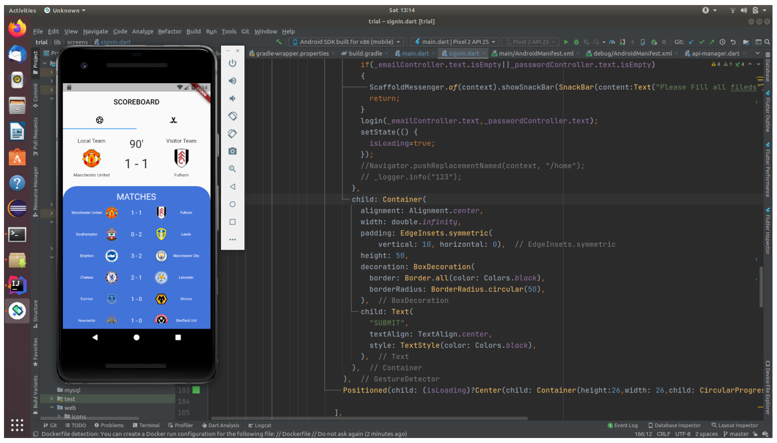
xyz
pandiavishal@gmail.com
12345678
User already registered

```
    if(_emailController.text.isEmpty||_passwordController.text.isEmpty)
    {
      ScaffoldMessenger.of(context).showSnackBar(SnackBar(content:Text("Please Fill all fileds"
      return;
    }
    login(_emailController.text,_passwordController.text);
    setState(() {
      isLoading=true;
    });
    //Navigator.pushReplacementNamed(context, "/home");
    // _logger.info("123");
  },
  child: Container(
    alignment: Alignment.center,
    width: double.infinity,
    padding: EdgeInsets.symmetric(
        vertical: 10, horizontal: 0),  // EdgeInsets.symmetric
    height: 50,
    decoration: BoxDecoration(
      border: Border.all(color: Colors.black),
      borderRadius: BorderRadius.circular(50),
    ),  // BoxDecoration
    child: Text(
      "SUBMIT",
      textAlign: TextAlign.center,
      style: TextStyle(color: Colors.black),
    ),  // Text
  ),  // Container
),  // GestureDetector
Positioned(child: (isLoading)?Center(child: Container(height:26,width: 26,child: CircularProgres

],
```

Figure 6: Example for Sign Up

3.3.3 Home Screen

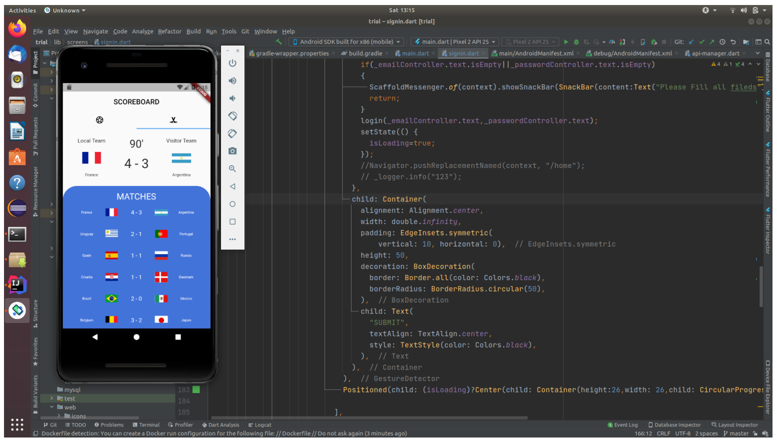
- Here we can see the current score of an ongoing match and the results of a match that has already been completed. [7]



```
      if(_emailController.text.isEmpty||_passwordController.text.isEmpty)
      {
        ScaffoldMessenger.of(context).showSnackBar(SnackBar(content:Text("Please Fill all fileds"
        return;
      }
      login(_emailController.text,_passwordController.text);
      setState(() {
        isLoading=true;
      });
      //Navigator.pushReplacementNamed(context, "/home");
      // _logger.info("123");
    },
    child: Container(
      alignment: Alignment.center,
      width: double.infinity,
      padding: EdgeInsets.symmetric(
          vertical: 10, horizontal: 0),  // EdgeInsets.symmetric
      height: 50,
      decoration: BoxDecoration(
        border: Border.all(color: Colors.black),
        borderRadius: BorderRadius.circular(50),
      ),  // BoxDecoration
      child: Text(
        "SUBMIT",
        textAlign: TextAlign.center,
        style: TextStyle(color: Colors.black),
      ),  // Text
    ),  // Container
  ),  // GestureDetector
  Positioned(child: (isLoading)?Center(child: Container(height:26,width: 26,child: CircularProgres

],
```

Figure 7: Home Screen

- Here we can switch between sports/tournaments to see different scores of the matches pertaining to that sport/tournament. [8]



```
      if(_emailController.text.isEmpty||_passwordController.text.isEmpty)
      {
        ScaffoldMessenger.of(context).showSnackBar(SnackBar(content:Text("Please Fill all fileds"
        return;
      }
      login(_emailController.text,_passwordController.text);
      setState(() {
        isLoading=true;
      });
      //Navigator.pushReplacementNamed(context, "/home");
      // _logger.info("123");
    },
    child: Container(
      alignment: Alignment.center,
      width: double.infinity,
      padding: EdgeInsets.symmetric(
          vertical: 10, horizontal: 0),  // EdgeInsets.symmetric
      height: 50,
      decoration: BoxDecoration(
        border: Border.all(color: Colors.black),
        borderRadius: BorderRadius.circular(50),
      ),  // BoxDecoration
      child: Text(
        "SUBMIT",
        textAlign: TextAlign.center,
        style: TextStyle(color: Colors.black),
      ),  // Text
    ),  // Container
  ),  // GestureDetector
  Positioned(child: (isLoading)?Center(child: Container(height:26,width: 26,child: CircularProgres

],
```

Figure 8: Switching between Sports

4 Software Development Life Cycle

4.1 Flutter - Frontend

Flutter is an open-source mobile SDK developer can use to build native-looking Android and iOS applications from the same code base. Widgets are the key of flutter, with widgets users build the entire UI. It provides developers with its own ready-made widgets that look native to Android or iOS apps, and also allows developers to make their own widgets.

The following are benefits we found while using Flutter:

- Flutter is a cross-platform application (same codebase for both andriod and iOs) which saves both time and money.
- Flutter offers outstanding performance due to its widgets and the Dart programming language.
- Flutter being open-source makes it easier to find help and necessary advice while developing an applicaiton.

The link here gives the instructions to install Flutter on your device.
SpandanApp has its core built from flutter and dart packages which are very easy-to-use and are very handy. SpandanApp comprises of three screens - **signup, signin, home**. Such a division was made entirely on the basis of the distinct functionalities that each of the screens offer. **Signup** screen offers the functionality of creating a new user whereas the **signin** screen provides the functionality to verify the login credentials entered and to identify a user as a verified user. Thirdly, the **home** screen provides the functionality of getting real-time updates of live matches and also to display the scorelines of recent matches that have taken place. To achieve this, SpandanApp uses a Restful API from an API provider API-Sports to get real-time data of the matches. An `api-manager.dart` was programmed to handle the API requests, to manage the API tokens & credentials, and to store the data received in a usable format.
Separate modules were programmed for different features present on the home screen, for e.g - `goalstat.dart` was programmed to display the live updates of an ongoing match along with features of displaying the timer, identifying the home team & away team. Similarly, `matchstat.dart` was written to handle the features of displaying the team names, team logos. `matchtile.dart` & `pagebody.dart` was used to display to scorelines of past matches in an attractive and appealing format.

4.2 PHP - Backend

PHP is an acronym for **Hypertext Preprocessor**. PHP is a server side scripting language that is embedded in HTML. It is used to manage dynamic content, databases, session tracking, even build entire e-commerce sites. In our project SpandanApp, we have used PHP as a backend tool to communicate between the **MySQL** database and the **flutter** app using GET, POST commands. To set-up an **un-interrupted connection** between the database and the app, a file named `mysqli_connect.php` has been written, to handle **login** verification requests, a separate `login.php` file has been written which basically takes in the inputs (e-mail & password) entered by the user and converts the input into JSON format and then checks the authenticity of the entered credentials by sending a query onto the MySQL database. Based on the response sent by the MySQL DB, the user is allowed or denied the access. Similarly for sign-up requests made on the app, a separate `registration.php` file is written which essentially inserts credentials and other profile parameters into the database. Below are the snippets of the mentioned files -

Listing 1: mysqli_connect.php

```php
<?php
require_once('./mysqli_connect.php');
$keys=array('email','password');
#$keys = file_get_contents('php://input');

for ($i = 0; $i < count($keys); $i++){
    if(!isset($_POST[$keys[$i]]))
    {
        $response['error'] = true;
        $response['message'] = 'Required Filed Missed';
        echo json_encode($response);
        return;
    }
}
#$obj = json_decode($json,true);
$email=$_POST['email'];
$password=$_POST['password'];
#echo $email;

$stmt = $con->prepare("SELECT * FROM user_registration WHERE
    email = ? AND password = ?");
  $stmt->bind_param("ss", $email, $password);
  $stmt->execute();
  $stmt->store_result();
  if($stmt->num_rows > 0)
  {
        $stmt->bind_result( $id, $name,  $email,$password);
    $stmt->fetch();

    $user = array(
    'id'=>$id,
    'name'=>$name,
    'password'=>$password,
    'email'=>$email
        );

    $stmt->close();
    $response['error'] = false;
    $response['message'] = 'User Logged In';
    $response['data'] = $user;
    }
    else
    {
        $response['error'] = true;
      $response['message'] = 'Invalid email or password';
      $stmt->close();
  }
echo json_encode($response);

?>
```

Listing 2: login.php

```
<?php
require_once('./mysqli_connect.php');
$keys=array('name','email','password');
//$keys=array('name','mobile','password','type');
for ($i = 0; $i < count($keys); $i++){
  if(!isset($_POST[$keys[$i]]))

   {
      $response['error'] = true;
      $response['message'] = 'Required Filed Missed';
      echo json_encode($response);
      return;
   }

}
$password=$_POST['password'];
$email=$_POST['email'];
$name=$_POST['name'];
//checking if the user is already exist with this email
//as the email should be unique for every user
```

```

if (isset($con)) {
    $stmt = $con->prepare("SELECT id FROM user_registration
   WHERE email = ? ");
}
$stmt->bind_param("s", $email);
$stmt->execute();
$stmt->store_result();

//if the user already exist in the database
if($stmt->num_rows > 0){
  $response['error'] = true;
  $response['message'] = 'User already registered';
  $stmt->close();

}
else
{
  #$user = array();
  //if user is new creating an insert query
  $stmt = $con->prepare("INSERT INTO user_registration ( name,
    email, password) VALUES (?,?,  ?)");
  $stmt->bind_param("sss",  $name, $email, $password);

  //if the user is successfully added to the database
  if($stmt->execute()){

    //fetching the user back
    $stmt = $con->prepare("SELECT * FROM user_registration
   WHERE email = ?");
    $stmt->bind_param("s",$email);
    $stmt->execute();
    $stmt->bind_result($id, $name, $email, $password);
    $stmt->fetch();

    $user=  array(
      'id'=>$id,
      'name'=>$name,
      'email'=>$email,
      'password'=>$password
        );

    $stmt->close();
        //adding the user data in response
    $response['error'] = false;
    $response['message'] = 'User registered successfully';
    $response['data'] = $user;
    }
}
```

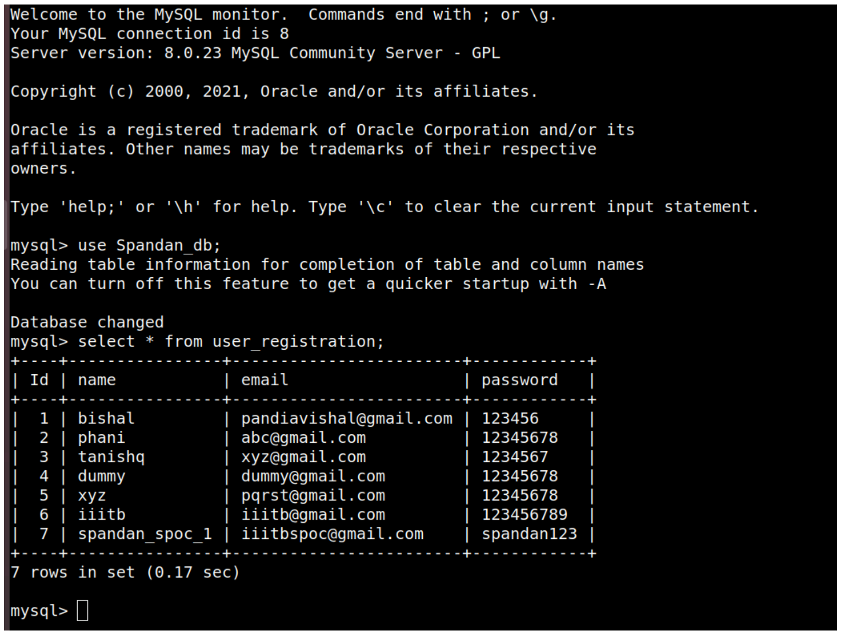
```
echo json_encode($response);
?>
```

Listing 3: registration.php

4.3 MySQL - Database

MySQL is a widely used open-source relational database management system (RDBMS). The MySQL development process focuses on offering a very efficient implementation of the features most people need. This means that MySQL still has fewer features than its chief open source competitor, PostgreSQL, or the commercial database engines.It uses a standard form of the well-known SQL data language. It works on many operating systems and works with many languages including PHP.

In our project SpandanApp, we have chosen MySQL as our data management tool because of its easy to use interface and keeping in mind its compatibility with our backend tool(PHP). Our database mainly comprises of a table named *user_registration* which essentially handles all the sign-up & sign-in requests. Below is the snapshot of the same -



```
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 8
Server version: 8.0.23 MySQL Community Server - GPL

Copyright (c) 2000, 2021, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> use Spandan_db;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> select * from user_registration;
+----+---------------+-----------------------+------------+
| Id | name          | email                 | password   |
+----+---------------+-----------------------+------------+
|  1 | bishal        | pandiavishal@gmail.com | 123456     |
|  2 | phani         | abc@gmail.com         | 12345678   |
|  3 | tanishq       | xyz@gmail.com         | 1234567    |
|  4 | dummy         | dummy@gmail.com       | 12345678   |
|  5 | xyz           | pqrst@gmail.com       | 12345678   |
|  6 | iiitb         | iiitb@gmail.com       | 123456789  |
|  7 | spandan_spoc_1 | iiitbspoc@gmail.com   | spandan123 |
+----+---------------+-----------------------+------------+
7 rows in set (0.17 sec)

mysql> 
```

Figure 9: User Registration table in MySQL DB

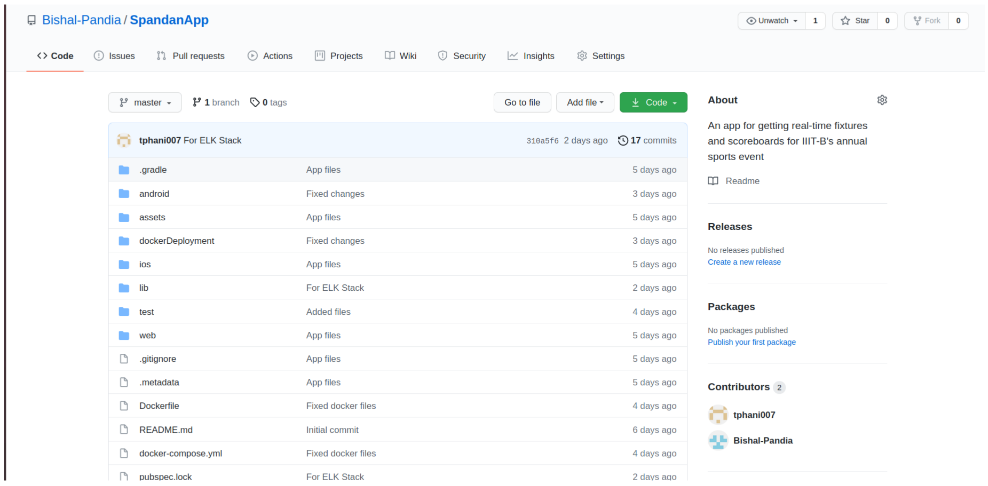
4.4 Git - Source Control Management

Git is a free and open source distributed version control system designed to handle everything from small to very large projects with speed and efficiency. Version control implies that it stores all versions and will come in handy for tracing back a particular version.

The following are the steps for creating a project on Git:

- First we create a repository on GitHub
- Next you have to pull the repository onto your local system
- Then you have to execute the following commands to push your code
  - git init
  - git add .
  - git commit -m "Commit Message"
  - git remote add origin *enter the url of the repository*
  - git push -u origin master

Once you follow the steps above, you will be to see the code up on GitHub. The following image shows the Github repository with all the artifacts.



tphani007 For ELK Stack 310a5f6 2 days ago 17 commits

.gradle	App files	5 days ago
android	Fixed changes	3 days ago
assets	App files	5 days ago
dockerDeployment	Fixed changes	3 days ago
ios	App files	5 days ago
lib	For ELK Stack	2 days ago
test	Added files	4 days ago
web	App files	5 days ago
.gitignore	App files	5 days ago
.metadata	App files	5 days ago
Dockerfile	Fixed docker files	4 days ago
README.md	Initial commit	6 days ago
docker-compose.yml	Fixed docker files	4 days ago
pubspec.lock	For ELK Stack	2 days ago

About

An app for getting real-time fixtures and scoreboards for IIIT-B's annual sports event

Readme

Releases

No releases published
Create a new release

Packages

No packages published
Publish your first package

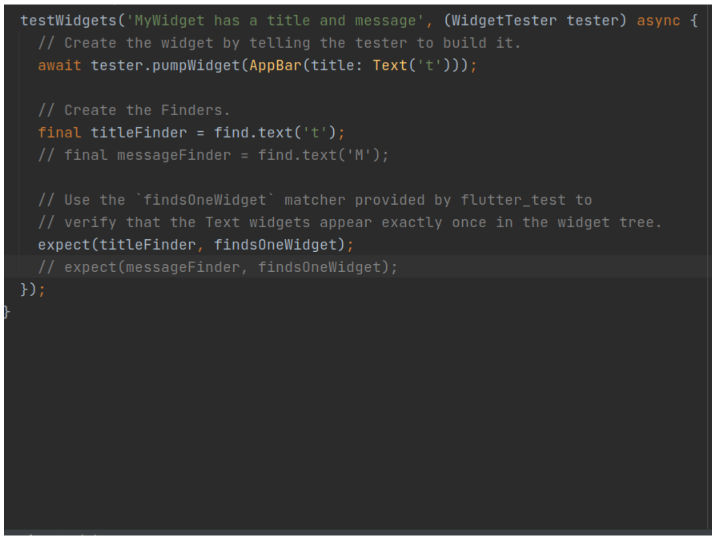
Contributors 2

tphani007
Bishal-Pandia

Figure 10: Spandan App Git repository

4.5 Flutter- Build & Testing

Flutter build was used to build the Apk with all the features and widgets. Before that the command *Flutter Doctor* was run to do a environment check and to check if all the packages and dependencies are installed or not to ensure that the build is smooth and uninterrupted. Flutter automatically creates test cases based on the widgets and action buttons that are being used in the app, or are part of the app. Based on this, test cases were automatically created in a file `test/widget_test.dart` and were then tested using *Flutter test*. Here is the snapshot of one such test case -



```
testWidgets('MyWidget has a title and message', (WidgetTester tester) async {
  // Create the widget by telling the tester to build it.
  await tester.pumpWidget(AppBar(title: Text('t')));

  // Create the Finders.
  final titleFinder = find.text('t');
  // final messageFinder = find.text('M');

  // Use the `findsOneWidget` matcher provided by flutter_test to
  // verify that the Text widgets appear exactly once in the widget tree.
  expect(titleFinder, findsOneWidget);
  // expect(messageFinder, findsOneWidget);
});
}
```

Figure 11: Test Case 1

4.6 Jenkins - Continuous Integration

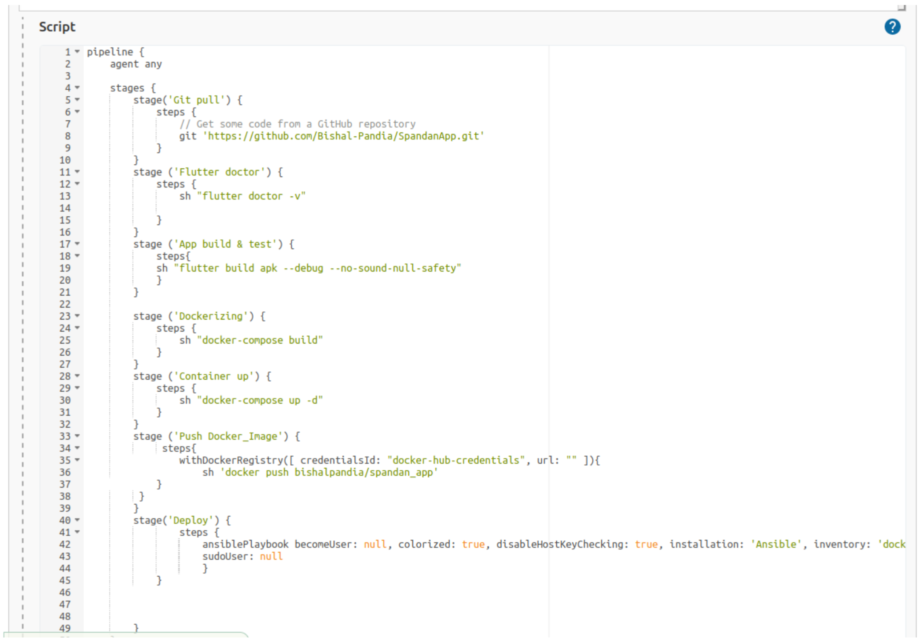
Continuous Integration is the most important part of DevOps that is used to integrate various DevOps stages. Jenkins is the most famous Continuous Integration tool. Jenkins makes it easier for us to integrate changes to the project, and making it easier for users to obtain a fresh build.

The link here can be used to install Jenkins onto you Ubuntu system. Once you are done with the installation and the set up, you can host the server in this link `https://localhost:8080/`

The next step is to create a pipeline on Jenkins. As soon as we create the pipeline on Jenkins the following are the stages in the pipeline:

- **Stage-1:** Git Pull
- **Stage-2:** Flutter Doctor
- **Stage-3:** Flutter Build & Test Apk
- **Stage-4:** Docker build
- **Stage-5:** Docker Compose
- **Stage-6:** Docker Push
- **Stage-7:** Ansible Deployment

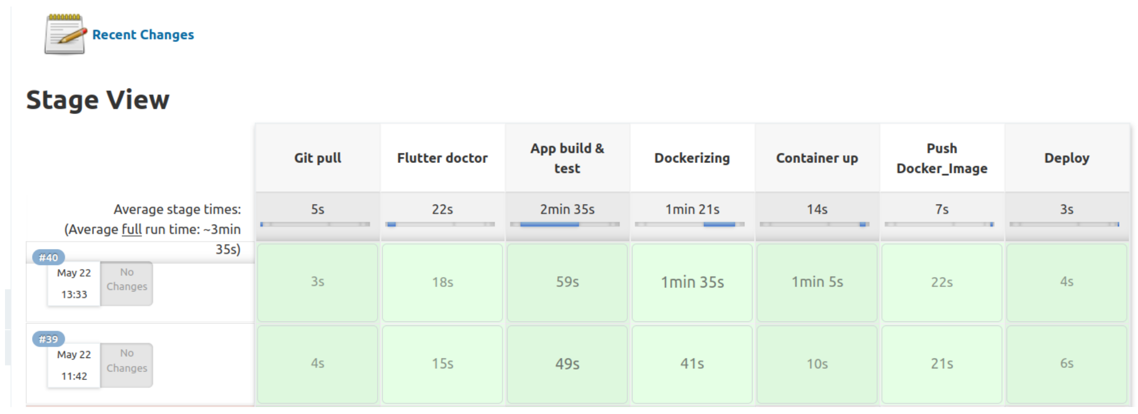
Here we can observer the script for the entire pipeline.



Script

```
pipeline {
    agent any

    stages {
        stage('Git pull') {
            steps {
                // Get some code from a GitHub repository
                git 'https://github.com/Bishal-Pandia/SpandanApp.git'
            }
        }
        stage ('Flutter doctor') {
            steps {
                sh "flutter doctor -v"

            }
        }
        stage ('App build & test') {
            steps{
            sh "flutter build apk --debug --no-sound-null-safety"
            }
        }

        stage ('Dockerizing') {
            steps {
                sh "docker-compose build"
            }
        }
        stage ('Container up') {
            steps {
                sh "docker-compose up -d"
            }
        }
        stage ('Push Docker_Image') {
             steps{
                withDockerRegistry([ credentialsId: "docker-hub-credentials", url: "" ]){
                    sh 'docker push bishalpandia/spandan_app'
            }
         }
        }
        stage('Deploy') {
                steps {
                    ansiblePlaybook becomeUser: null, colorized: true, disableHostKeyChecking: true, installation: 'Ansible', inventory: 'dock
                    sudoUser: null
                    }
            }



        }
```

Figure 12: Jenkins pipeline script


Recent Changes

Stage View

	Git pull	Flutter doctor	App build & test	Dockerizing	Container up	Push Docker_Image	Deploy
Average stage times: (Average full run time: ~3min 35s)	5s	22s	2min 35s	1min 21s	14s	7s	3s
#40 May 22 13:33 No Changes	3s	18s	59s	1min 35s	1min 5s	22s	4s
#39 May 22 11:42 No Changes	4s	15s	49s	41s	10s	21s	6s

Figure 13: Jenkins pipeline status

4.7 Docker - Containerization

Docker is a tool designed to make it easier to create, deploy, and run applications by using containers. Containers allow us to package up an application with all of the parts it needs, such as libraries and other dependencies, and deploy it as one package. The docker images can be pushed onto the DockerHub and can be made public so that anyone can pull the image and then get the application running on their system.

The following are the steps to install docker:

- sudo apt install apt-transport-https ca-certificates curl software-properties-common
- curl -fsSL https://get.docker.com -o get-socker.sh
- sh get-docker.sh

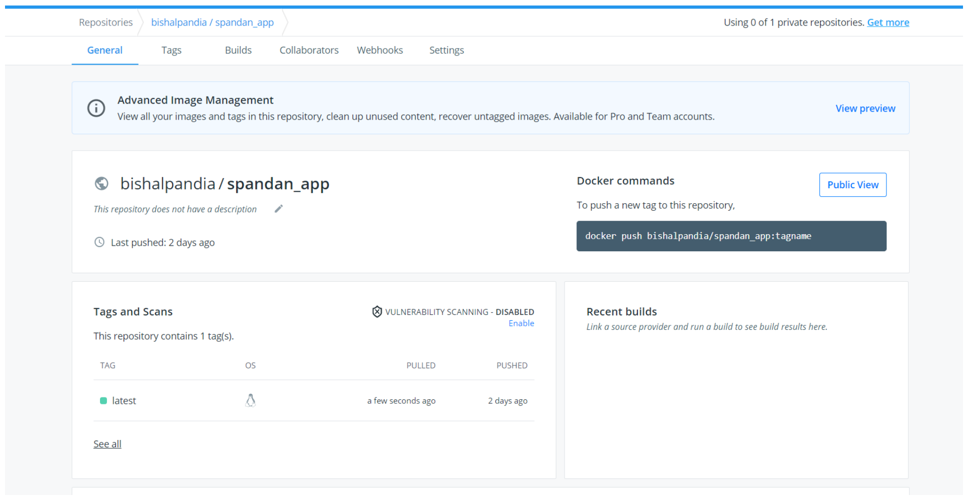
SpandanApp uses docker-compose to build the docker containers and also to setup a network between the containers for an easy exchange of data. SpandanApp comprises of two docker containers - A container with the Apk along with PHP & a container for the MySQL database. To achieve this, firstly, a `Dockerfile` was written to specify the tools and packages required in building the image of the apk along with PHP support and secondly a `docker-compose.yml` was written to pull the image of **MySQL** from the DockerHub and to build the image of the apk using the specifications defined in the `Dockerfile`. `docker-compose.yml` was also used to setup a network between the two services- database and the web service by specifying the port mappings, docker volumes were defined in order to persist the data and to persist other necessary configuration files. Below are the code snippets of `Dockerfile` & `docker-compose.yml`-

```
version: '3'
services:
  mysql:
    image: mysql:5.7
    volumes:
      - ./mysql:/docker-entrypoint-initdb.d
    restart: always
    environment:
      - MYSQL_ROOT_PASSWORD=root
      - MYSQL_DATABASE=Spandan_db
    ports:
      - "4000:3306"
  web:
    build: .
    volumes:
      - .:/var/www/html
    ports:
      - "4001:80"
    depends_on:
      - mysql
    stdin_open: true
    tty: true
```

Listing 1: docker-compose.yml

```
FROM ubuntu:18.04
MAINTAINER bishalpandia bishalpandia@gmail.com
#Prepare environment with necessary dependencies
ARG DEBIAN_FRONTEND=noninteractive
RUN apt-get update && apt-get install -y curl git unzip xz-
    utils zip libglu1-mesa openjdk-8-jdk wget
RUN useradd -ms /bin/bash developer

USER root0nc
WORKDIR ./
RUN apt-get install -y php

```

```
# Prepare Android directories and system variables
RUN mkdir -p Android/sdk
ENV ANDROID_SDK_ROOT ./Android/sdk
RUN mkdir -p .android && touch .android/repositories.cfg

# Set up Android SDK
RUN wget -O sdk-tools.zip https://dl.google.com/android/
    repository/sdk-tools-linux-4333796.zip
RUN unzip sdk-tools.zip && rm sdk-tools.zip
RUN mv tools Android/sdk/tools
RUN cd Android/sdk/tools/bin && yes | ./sdkmanager --licenses
RUN cd Android/sdk/tools/bin && ./sdkmanager "build-tools
    ;29.0.2" "patcher;v4" "platform-tools" "platforms;android
    -29" "sources;android-29"
ENV PATH "$PATH:./Android/sdk/platform-tools"

# Download Flutter SDK
RUN git clone https://github.com/flutter/flutter.git
ENV PATH "$PATH:./flutter/bin"

# Run basic check to download Dart SDK
RUN flutter doctor
RUN git clone https://github.com/Bishal-Pandia/SpandanApp.git
```
Listing 4: Dockerfile



Advanced Image Management
View all your images and tags in this repository, clean up unused content, recover untagged images. Available for Pro and Team accounts.

bishalpandia / spandan_app
This repository does not have a description
Last pushed: 2 days ago

Docker commands
To push a new tag to this repository,
```
docker push bishalpandia/spandan_app:tagname
```

Tags and Scans
VULNERABILITY SCANNING - DISABLED
This repository contains 1 tag(s).

TAG	OS	PULLED	PUSHED
latest	🐧	a few seconds ago	2 days ago

Recent builds
Link a source provider and run a build to see build results here.

Figure 14: Dockerhub Repository

4.8 Ansible - Continuous Deployment

Ansible is a configuration management tool, which is the process of maintaining computer systems, servers, and software in a desired, consistent state. Continuous deployment is the process of automating the process of deploying applications to multiple client machines. Ansible can be used for continuous deployment by pushing the required files to the managed hosts. Ansible runs on the control node which works as the central node for distributing files to the managed host.

The following link can be used to check how to install Ansible onto your device.

In the SDLC, for the deploy stage, the following code is for the `deployDocker.yml`[2] file, which is the playbook and the corresponding inventory file[3] as well.

```yaml
---
- name: Pull Docker image of spandan_app
  hosts: local
  connection: local
  vars:
    ansible_python_interpreter: /usr/bin/python
  tasks:
    - name: Pull image spandan_app
      docker_image:
        name: bishalpandia/spandan_app:latest
        source: pull
```

Listing 2: deployDocker.yml

```
localhost ansible_user=bishal
```

Listing 3: inventory file

4.9 ELK Stack - Continuous Monitoring

Continuous monitoring is an automated process by which DevOps personnel can observe and detect compliance issues and security threats during each phase of the DevOps pipeline. Continuous Monitoring comes in at the end of the DevOps pipeline. Once the software is released into production, Continuous Monitoring will notify us in the event of specific issues arising in the prod environment. It provides feedback on what is going wrong, which allows the relevant people to work on necessary fixes as soon as possible.

ELK stands for:

- ElasticSearch: This is a search engine that takes the input from Logstash and shifts through the pre-processed logs.
- Logstash: Collects logs from various sources and user filters like grok to parse the logs and organizes them in fields defined by us.
- Kibana: Kibana is a visualization engine that runs on top of Elasticsearch. This is used to visualize various aspects of the logs and gain insights of that help understand the logs even better. [15]

For the ELK stack we have used the assistance of this website that allows us to monitor and use the ELK cloud services. The reason for this is because logging support for Flutter applications was quite difficult to find an integrate and this software makes it easier to integrate the logging process from Flutter to ELK stack. The packages that were used for logging information are

- logging 1.0.1: This helped creating the logs that are to be generated in the app.
- logging_appenders 1.0.0: This helped in sending the logs to the website which provided an ELK stack platform.

The logs[16] that are generated in the app are based on the following:

- When a user signs into the app
- When the user creates an account
- When the user switches from login page to sign up page

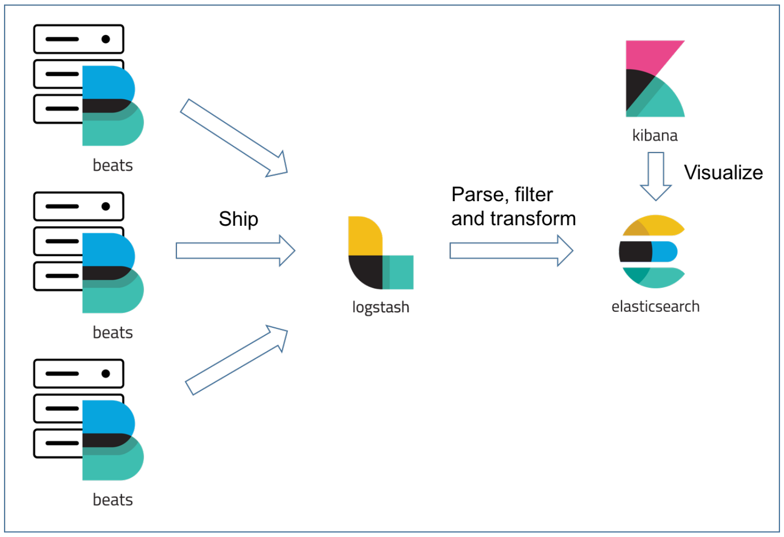
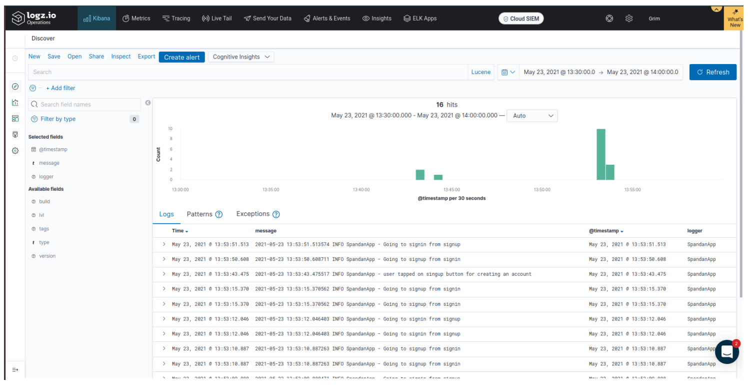


Figure 15: ELK Architecture



May 23, 2021 @ 13:30:00.0 → May 23, 2021 @ 14:00:00.0

16 hits

May 23, 2021 @ 13:30:00.000 - May 23, 2021 @ 14:00:00.000

```
Time                         message                                                                                   @timestamp                   logger
May 23, 2021 @ 13:53:51.513  2021-05-23 13:53:51.513574 INFO SpandanApp - Going to signin from signup                   May 23, 2021 @ 13:53:51.513  SpandanApp
May 23, 2021 @ 13:53:50.608  2021-05-23 13:53:50.608711 INFO SpandanApp - Going to signup from signin                   May 23, 2021 @ 13:53:50.608  SpandanApp
May 23, 2021 @ 13:53:43.475  2021-05-23 13:53:43.475517 INFO SpandanApp - user tapped on singup button for creating an account  May 23, 2021 @ 13:53:43.475  SpandanApp
May 23, 2021 @ 13:53:15.370  2021-05-23 13:53:15.370562 INFO SpandanApp - Going to signup from signin                   May 23, 2021 @ 13:53:15.370  SpandanApp
May 23, 2021 @ 13:53:15.370  2021-05-23 13:53:15.370562 INFO SpandanApp - Going to signup from signin                   May 23, 2021 @ 13:53:15.370  SpandanApp
May 23, 2021 @ 13:53:12.046  2021-05-23 13:53:12.046403 INFO SpandanApp - Going to signin from signup                   May 23, 2021 @ 13:53:12.046  SpandanApp
May 23, 2021 @ 13:53:12.046  2021-05-23 13:53:12.046403 INFO SpandanApp - Going to signin from signup                   May 23, 2021 @ 13:53:12.046  SpandanApp
May 23, 2021 @ 13:53:10.887  2021-05-23 13:53:10.887263 INFO SpandanApp - Going to signup from signin                   May 23, 2021 @ 13:53:10.887  SpandanApp
May 23, 2021 @ 13:53:10.887  2021-05-23 13:53:10.887263 INFO SpandanApp - Going to signup from signin                   May 23, 2021 @ 13:53:10.887  SpandanApp
```

Figure 16: Kibana

5 Future Scope

- We plan on implementing a notification system, where user who are interested in a particular team/match can check out any information and updates regarding their matches.
- We plan on implementing a team viewing system, where users can view their favorite teams/players and get information regarding the team's status and fixtures.
- We plan on creating an Admin user that has access to modifying the fixtures, user information and also the sports and tournament that are to be conducted.

6 Conclusion

We have a learnt a lot of aspects as to what goes on into developing an application. Using DevOps tools made us realise how easy things become when it comes to releasing a new feature or the integrating the operations part with the development as well. Overall this project helped us in understanding the importance of DevOps and how the different stages have their own advantages and play a crucial role in the entire process. The link to the GitHub Repository and the DockerHub Repository can be found here.

7 References

- https://docs.docker.com/compose/gettingstarted/
- https://www.digitalocean.com/community/tutorials/how-to-install-jenkins-on-ubuntu-18-04
- https://pub.dev/packages/logger
- https://pub.dev/packages/logging_appenders
- https://flutter.dev/docs/get-started/install
- https://docs.ansible.com/ansible/latest/user_guide/playbooks_intro.html
- https://medium.com/saas-startup-factory/remote-logging-with-flutter-on-the-logz-io-elk-stack-f360df143ef0
- https://www.php.net/manual/en/book.mysql.php
- https://flutter.dev/docs/testing
- https://flutter.dev/docs/cookbook/testing/unit/introduction
- https://flutter.dev/docs/deployment/android