

# TEST PLAN

## Introduction

- We have implemented The Agile Model to test our project.
  - In software development, Agile practices involve discovering requirements and developing solutions through the collaborative effort of self-organizing and cross-functional teams and their end users.
  - It advocates adaptive planning, evolutionary development, early delivery, and continual improvement, and it encourages flexible responses to change.
- 

## Quality Objective

HeartDoc aims to achieve these objectives requiring three key things in performance testing: usability, performance, and stability.

1. **Usability**- how the user interacts with your mobile application. If they can't operate your app, they won't keep it. Testing our application for usability for these considerations:-
  2. **Accessibility** – Our app should allow users to access their data when they want it. Also HeartDoc should be compatible with all the devices and operating systems.
  3. **Functionality** – Manual Testing has been done to check all the aspects of the app's usability throughout the development and testing process, checking all the functionality on real devices on a regular basis.
-

# Test Methodology

## Overview:

1. The Agile Model advocates adaptive planning, evolutionary development, early delivery, and continual improvement, and it encourages flexible responses to change.
2. We had a detailed analysis on the design, workflow and requirements of our project and hence we had decided beforehand which model to choose.
3. There were some bugs which we have debugged successfully and are looking forward to extending our implementation levels further to increase the reachability and efficiency of our product.

## Test Levels:

1. **Unit Test:** In this level we will test each individual component for example, we will check the Student, FnA and Hod.
  2. **Integration Test:** In this level the integrated components will be tested. Incremental testing is used in agile development methods and hence, every release of the project is tested thoroughly. This ensures that any bugs in the system are fixed before the next release.
  3. **System Test:** In this level, the entire system will be tested against different tests and checking whether any bug arises.
  4. **Acceptance test:** In this level we will ask different end users, to check the app for themselves, these people will check the efficiency and practicality of the app and hence we determine the reachability of our app.
- 

# Test Deliverables

## Test Cases:

- TC01:-LOGIN
- TC02:-PAY FEE

- TC03:-NOTIFY
  - TC04:-ALLOTDATE\_COURSE :
  - TC05:-UPDATE\_TIMETABLE
  - TC06:-UPDATE\_NOTICEBOARD
  - TC07:-CHANGE\_PASSWORD
  - TC08:-UPDATE\_PROFILEPICTURE
  - TC09:-UPLOAD\_FEECHART
  - TC10:-SEND\_FEELAERT
  - TC11:-LATEFEE\_TIME\_EXTENSION
  - TC12:-LATE\_FEE\_AMOUN\_TCHANGE
  - TC13:-APPLY\_LATEFEE
  - TC14:-REGISTER\_NEW\_STUDENT
  - TC15:-APPROVE\_FEES
  - TC16:-ADD\_EVENT
  - TC17:-ASSIGN\_FACULTY
  - TC18:-ADD\_NEW\_COURSE
  - TC19:-UPDATE\_COURSE
  - TC20:-DELETE\_COURSE
  - TC21:-UPDATE\_EXAM\_SCHEDULE
  - TC22:-LOGOUT
-

# Modules

Module_id	Module
Mod1	Login
Mod2	HoD Section
Mod3	FnA Section
Mod4	Student Section

---

# Requirements

<u>R_ID</u>	<u>Requirements</u>
R1	Login
R2	Calendar Display
R3	Faculty List Management (HOD)
R4	Course Modification (HOD)
R5	Time Table Display
R6	Exam Schedule Display

R7	Notification
R8	Manage Fees Chart (FnA)
R9	Defaulters List Display
R10	Fee Approval
R11	Check Fee Status

---

# Methods

M_ID	Methods
M1, M2	input_cred(), verification()
M3, M4	input_updated_values, update()
M5	login_type
M6	notification()
M7	display()
M9	fee_approval()
M10	view_profile()
M11	course_allotment()
M12	course_modification()
M13	manage_fee_chart()

---

# TRACEABILITY MATRIX

REQUIREMENTS	USE-CASE	HIGH-LEVEL DESIGN ELEMENT	LOW LEVEL DESIGN ELEMENT
R1	UC-1	Mod1	M1. M2
R2	UC-2	Mod4	M7
R3	UC-3	Mod2	M10, M12
R4	UC-4	Mod2	M12
R5	UC-7	Mod4	M7
R6	UC-8	Mod4	M7
R7	UC-11	Mod2, Mod3, Mod4	M6
R8	UC-13	Mod3	M13
R9	UC-16	Mod3	M7
R10	UC-15	Mod3	M9
R11	UC-14	Mod3, Mod4	M7, M10

---

# Test Plan

FUNCTION	INPUT	EXPECTED OUTPUT	GENERATED OUTPUT
<b>LOGIN :</b> <b>USERNAME = ABC</b> <b>PASSWORD = 1234</b> <b>CLICK ON LOGIN</b> <b>BUTTON</b>	1) USERNAME = "ABC" PASSWORD = "123" 2) USERNAME = "ABCD" PASSWORD = "1234" 3) USERNAME = "ABC" PASSWORD = "1234"	1) "PLEASE ENTER VALID CREDENTIALS" 2) "PLEASE ENTER VALID CREDENTIALS" 3) "LOGIN SUCCESSFUL"	1) "PLEASE ENTER VALID CREDENTIALS" 2) "PLEASE ENTER VALID CREDENTIALS" 3) "LOGIN SUCCESSFULL"
<b>PAYFEE :</b> <b>AMOUNT = REQ;</b> <b>CLICK ON PAY BUTTON</b>	1) AMOUNT = REQ(BY DEFAULT) PAY	1) FEE PAYMENT SUCCESS	1) FEE PAYMENT SUCCESS
<b>NOTIFY :</b> <b>TEXT FIELD = "REQUIRED NOTIFICATION"</b> <b>CLICK ON SEND BUTTON</b>	1) TEXT FIELD = "TEST NOTIFY" 2) TEXT FIELD = "TEST NOTIFY" 3) TEXT FIELD = "TEST NOTIFY"	1) "NOTIFICATION SENT SUCCESSFULLY" 2) "NOTIFICATION SENT SUCCESSFULLY" 3) "NOTIFICATION SENT SUCCESSFULLY"	1) "NOTIFICATION SENT SUCCESSFULLY" 2) "NOTIFICATION SENT SUCCESSFULLY" 3) "NOTIFICATION SENT SUCCESSFULLY"
<b>ALLOTDATE_COURSE :</b> <b>COORDINATOR = "";</b> <b>COURSE="";</b> <b>CODE="";</b>	1) COORDINATOR = "Triloki Pant" COURSE = "PPL" CODE = "PPL"	1) "COURSE ALLOCATES SUCCESSFULLY" (with change in database)	1) "COURSE ALLOCATED SUCCESSFULLY" (with change in database)
<b>UPDATE_TIMETABLE</b> <b>CLICK ON UPLOAD BUTTON</b>	1) Upload new time table	1) Change in time table	1) Time Table Changed

<b>UPDATE_NOTICEBOARD CLICK ON UPDATE BUTTON</b>	1)	Upload new notice board	1)	Change in notice board	1)	Change in notice board
<b>CHANGE_PASSWORD CLICK ON CHANGE PASSWORD BUTTON IN STUDENT'S PORTAL</b>	1) 2)	Enter present password Enter new password	1)	Change of password in database	1)	Change of password in database
<b>UPDATE_PROFILEPICTURE CLICK ON UPDATE BUTTON</b>	1) 2)	Select new image Upload the image	1)	Upload of new profile picture	1)	Upload of new profile picture
<b>UPLOAD_FEECHART CLICK ON UPLOAD BUTTON</b>	1)	Select new fee chart	1)	Replacement of fee chart	1)	Replacement of fee chart
<b>SEND_FEELERT CLICK ON ALERT BUTTON</b>	1)	Click on the send alert button	1)	Alert sent to defaulters	1)	Alert sent to defaulters
<b>LATEFEE_TIME_EXTENSION CLICK ON CHANGE DATE BUTTON</b>	1)	Date Entered = '27-05-2021'	1)	Date extended for fee payment till '27-05-2021'	1)	Date extended for fee payment till '27-05-2021'
<b>LATE_FEE_AMOUNT_CHANGE CLICK ON CHANGE AMOUNT BUTTON</b>	1)	Entered Amount = '200'	1)	Amount Changed for late fee to 200.	1)	Amount Changed for late fee to 200.
<b>APPLY_LATEFEE CLICK ON APPLY LATE FEE BUTTON</b>	1)	Click on Apply late fees	1)	Late fees added up to the previous value of late fees for all students with due>0.	1)	Late fees added up to the previous value of late fees for all students with due>0.
<b>REGISTER_NEW_STUDENT CLICK ON REGISTER BUTTON</b>	1) 2)	Enter Student info with enrollment number Assigned previously. Enter Student info with enrollment number that is not Assigned previously.	1) 2)	It won't move to next step and no new student will be registered. It moves to next step and finally a new student is added.	1) 2)	It won't move to next step and no new student will be registered. It moves to next step and finally a new student is added.



<b>APPROVE_FEES CLICK ON APPROVE BUTTON</b>	1) Clicked on Approve button	1) Fee Approval request disappears from the Fee Approval list and mark feeApproval attribute for that Student true.	1) Fee Approval request disappears from the Fee Approval list and mark feeApproval attribute for that Student true.
<b>ADD_EVENT CLICK ON ADD BUTTON</b>	1) Select Start and Ending date and set title to 'Sample Event'	1) Event gets added to DB and could be seen in the Calenders.	1) Event gets added to DB and could be seen in the Calenders.
<b>ASSIGN_FACULTY CLICK ON + BUTTON and then CLICK ON COURSE</b>	1) CLICK ON + BUTTON in front of faculty name and then click in the course to be allotted to.	1) New Assignment of teacher to course added in the table.	1) New Assignment of teacher to course added in the table.
<b>ADD_NEW_COURSE CLICK ON ADD BUTTON</b>	1) Add data for new course to be added and click on Add.	2) New Course with entered data added to the list.	2) New Course with entered data added to the list.
<b>UPDATE_COURSE CLICK ON UPDATE BUTTON</b>	1) Select the course to be updated. Then change the data and click on update.	3) The updated information reflects in the list,	3) The updated information reflects in the list,
<b>DELETE_COURSE CLICK ON DELETE BUTTON</b>	1) Click On delete button for the particular course.	4) Course is deleted from the list.	4) Course is deleted from the list.
<b>UPDATE_EXAM_SCHEDULE CLICK ON UPLOAD BUTTON</b>	1) Upload new exam schedule.	5) Change in exam schedule	5) Change in exam schedule
<b>LOGOUT CLICK ON LOGOUT BUTTON</b>	1) Click on the logout button	1) Takes you back to login page.	1) Takes you back to login page.