

INTERNSHIP REPORT ON

TASC IOS APP

(Feb 2022-June 2022)

**Internship report submitted in partial fulfillment of the requirement for the degree of
Bachelor of Technology in**

Computer Science and Engineering



BY

ABHISHEK GUPTA ENROLL NO: 181243

Department of Computer Science & Engineering and Information Technology

**Jaypee University of Information Technology, Waknaghat, 173234, Himachal Pradesh,
INDIA**

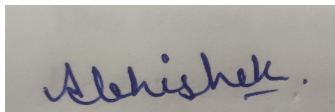
TABLE OF CONTENT	
Content	Page No.
Declaration By Candidate	I
Acknowledgement	II
Certificate	III
List of Figures	IV
Chapter 01: INTRODUCTION	1
1.1 Introduction	1
1.2 Job description	2
1.3 Methodology	3
1.4 iOS development requirement	3
1.5 Problem Statement	5
1.6 Objective	6
Chapter 02: COMPANY DESCRIPTION	7
2.1 ZopSmart Technologies	9
2.2 ZopSmart Major Clients	12
Chapter 03: LITERATURE SURVEY	13
Chapter 04: TOOLS AND TECHN.	19
3.2 SWIFTUI	19
3.3 Github	20
3.4 Charles	20
Chapter 05: LIVE PROJECT	21
4.1 Tasc App	25
4.2 System Design	35
Chapter 06: CONCLUSION	41
5.1 Conclusion	42
5.2 Future Scope	42

5.2 Mentor's Review	43
REFERENCES	45
PLAGIARISM REPORT	46

DECLARATION

I hereby declare that this submission is my own work carried out at Zopsmart Technologies Pvt Ltd, Bangalore from February 2022 to June 2022 and that, to the best of my knowledge and belief, it contains no material previously published or written by another person nor material which has been accepted for the award of any other degree or diploma from a university or other institute of higher learning, except where due acknowledgment has been made in the text.

SUBMITTED BY:

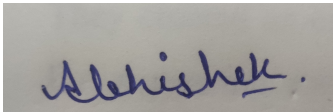
A rectangular box containing a handwritten signature in blue ink that reads "Abhishek".

Abhishek Gupta 181243

Computer Science & Engineering and Information Technology Department. Jaypee
University of Information Technology, Wahnaghat, Solan

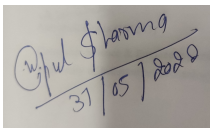
CERTIFICATE

This is to certify that the work which is being presented in the project report titled “**TASC iOS Apps**” in the fulfillment of the requirements for the award of the degree of B.Tech in Computer Science And Engineering and submitted to the Department of Computer Science And Engineering, Jaypee University of Information Technology, Waknaghat, Solan is an authentic record of work carried out by **Abhishek Gupta (181243)** under the supervision of Mr. Narendran Anil (Engineering Manager), Zopsmart Technologies, Bangalore, India.



ABHISHEK GUPTA
181243

The above statement made is correct to the best of my knowledge.



ACKNOWLEDGEMENT

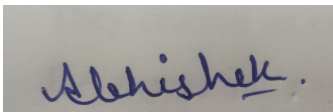
I would like to thank Rashmi Singh, Manager of HR in Talent Acquisition, of Zopsmart Technologies, Bangalore for allowing me to do an internship within the organization.

I also would like to thank Mr. Pratyush Kumar, Mr. Narendran Anil, and all the people that worked along with me at Zopsmart Technologies, Bangalore for their patience and openness. They created an enjoyable working environment. I also would like to thank Ayush Tiwari for mentoring me throughout my internship and helping me learn new concepts and technologies used in iOS development.

It is indeed with a great sense of pleasure and immense sense of gratitude that I acknowledge the help of these individuals.

I am highly indebted to Mr. Pankaj Kumar, Training & Placement Coordinator of our college for the facilities provided to accomplish this internship. I would also like to thank the Head of our Department Dr.Vivek Kumar Sehgal and the faculty for teaching us the skills required for this internship. And special thanks to my mentor in the college Vipul Kumar Sharma who supported me throughout the whole and guided me to achieve the best.

Finally, I must acknowledge the constant support and patients of my parents.



Abhishek Gupta (181243)

Computer Science & Engineering and Information Technology Department. Jaypee
University of Information Technology, Wagnaghat, Solan

LIST OF FIGURES

Fig no.	Figures	Page no.
1	Software development cycle	1
2	iOS development cycle	2
3	ZopSmart Logo	5
4	Swift Language Logo	10
5	New Project Template Screen	11
6	New Project Targets Screen	11
7	New Xcode Project View	12
8	Xcode project with multiple views and controllers	13
9	Xcode project with Launch Screen	13
10	Charles screen showing API calls.	15
11	Charles showing API Summary	15
12	Create Task Screen showing available details	17
13	Create task Screen showing prefilled details	17
14	Create Task Screen Item List view	18
15	Team Screen	18
16	Coupons popup view Controller	19
17	Add Task popup view Controller	19
18	Settings Screen	20
19	Add Team members popupView	20
20	Reports View Controller	21
21	Notifications Screen	21
22	Settings Popup view	22

CH- 01 - INTRODUCTION

1.1 Introduction

The application made is basically the todo list app which helps in the management of different tasks. Allow your users to swiftly add new activities to Application using our very user friendly method by utilising our Quick Add endpoint. You may assist your customers find material within your app faster and return more frequently by automatically attaching links to your app to the task content. Quick Add is the quickest way to get started using the Api which are made using GoLang technology.

We can add different information along with the title and description, the main thing we can add the due date of the task, next we can add the priority of the task i.e how important is that task. Along with this we can assign it to some individual or particular team.

The dueDate has an important characteristic i.e we get a notification when the due time of the task remaining is 5 minutes. And also there is an Api made for the notification i.e when any user get assigned any task or his team get assigned any task, the user will get the notifications. Several features are there in this app like the login setup is proper there, the old user can simply log in to the application and the new user can make the new account.

Also there is also support for the forgot password, when the user forgets his or her credentials, he or she can reset the password by simply sending the request to particular email id , once user clicks on reset password, after entering the email-id, the user will get redirected to the login page and default password is sent to that email-id. Along with this the change password functionality is also there, i.e user can create the password of his/her choice by using the current password, with only one condition that the password should be atleast 6 characters long.

The most important feature of the application are repeating task, i.e the task can be made repeating i.e on daily, weekly, monthly, and yearly basis. The user can select the start, end date and particular days of week or particular date, when the user wants that particular task to be made repeating.

Next any user can comment on task, the comment of that task is visible in the task activity screen, once commented, the user can edit that comment.

Also we can edit the task which are created at any time, the changes made to the task from its level of being created to current time, all the editing history is visible in task activity screen.

After tapping on any particular task, user is redirected to task activity screen.

This task has an important characteristic of having as much subtasks, there is no limit of number of subtasks of any particular task, all the subtask has one field called parentId, which is same for all the subtasks which is equal to the taskId of the parent task.

Every subtask can have weight or not, depending on user adds the weight to that subtask, as much number of subtasks are completed, there total weight is calculated and that number is equal to that parent task progress.

This subtask has all the same features and fields as one having task with only one condition i.e all the subtask has same parentId which is equal to the taskId of the task.

We have the concept of teams i.e user can create the team and he can add the members to the team by only knowing their email-id, the default owner of the team is the user itself, the owner has the right to edit and delete the team and members has the right to leave the team.

Various snackbars are added at the appropriate places, i.e after deleting the task, subtask, deleting the team and removing the member from the team of which only the owner has the right.

A snackbar is visible for 5 seconds, with undo button on it, if the undo button is pressed, that action will be reversed, if the user doesn't press that undo button, the snackbar will just simply dismiss after 5 seconds via completing that action.

Image upload feature is also there in the user profile and team image, we can upload the image of our choice, if any particular hasn't uploaded his or her image or the team doesn't has any image, the app will take image from the default gravatar instead of not showing any image.

Notifications column is there in which the user gets all the list of notifications till now, the notifications which the user has seen had different colour and the notifications which are not seen by the user till now had a different background color.

A filter view is there in the homescreen, which contains different features like priority tasks, overdue tasks, upcoming tasks, my tasks etc. When the user clicks on any particular filter, the homescreen is loaded with all the tasks after applying that particular filter.

A settings column is made which contains different checksums depending upon receiving the notifications like receiving the team notifications, individual notifications etc.

The application name is tasc which is available on the AppStore for android and the authorization for ios is under progress, once the ios app is approved by Apple, IOS app will be live on AppStore.

1.2 Problem Statement

We find it challenging to keep focused on our tasks and complete our work efficiently when we are distracted by social media and other conveniently accessible online diversions.

Furthermore, moving between tasks frequently may give us the misleading impression that we are productive when we are not. Rather than focusing on erasing tiny items from our todo list only for the sake of looks, we should prioritise chores and work on the most critical ones.

This app's objective is to assist us become more conscious of how we spend time when performing things and how productive that time is.

It can help us put some limits on social media to prevent distraction and keep track of how much time we spend working on our to-do list. We can better organise our daily routines when we have a better understanding of the expected time we'll need to spend on our duties, as well as the validated time spent on the things for reference or personal/team reviews.

Basically in today's busy world, it is very difficult to manage the daily task. This app is basically the solution to it. We can customize it according to the user need, along with different features and functionalities. The dueDate and time plays an important functionality which notifies the user when the date and time is get to over.

Also this application also indicates the task which are completed and which are pending.

1.3 Objective

Everybody has a list of things to accomplish.

You can organise your daily living routine with the to-do list. If not, make a list of goals you want to achieve in the next year or so.

To-do lists can help you organise your life routines as well as uncover unfulfilled desires. It is not always possible to have a pen and paper with you wherever you go in order to write down your desires or to-do list.

For your convenience, we'll speak about some of the finest apps for to-do lists in this piece, but first, let's explain what a to-do list is and why it's important.

If a person makes to-do lists, he will almost certainly stick to them without fail.

It's the most effective technique to keep yourself and your job organised. We can study the fundamentals and practises of making a to-do list in the apps.

By regularly alarming us, it will help us become more organised and fulfil our responsibilities. To-do lists help us decide and spend our time precisely by balancing work and other goals.

The objective of this app is to help the user to manage his daily routine task, because in this busy world with lot of daily works to do , it is impossible for the user to manage the tasks by remembering them all and completing before the deadline. This app does the same for the user, we do not have to remember the task, the app just notifies at the appropriate time and you can add the task, edit that task with lot of different features.

These kind of to-dolist helps the user in managing their routine tasks and also helps them to not miss any particular task and in these time where every person has a smartphone, downloading such application is not a big task which helps the user to manage their tasks. This app's objective is to assist us become more conscious of how we spend time when performing things and how productive that time is.

It can help us put some limits on social media to prevent distraction and keep track of how much time we spend working on our to-do list. We can better organise our daily routines when we have a better understanding of the expected time we'll need to spend on our duties, as well as the validated time spent on the things for reference or personal/team reviews.

1.4 Methodology

Since we develop for iOS, we work on mac environment. XCode is the tool we used . And swift is the language used here . For the UI , we had two options in the company: UI Kit and SwiftUI.

UIKit has been used for a long time . SwiftUI is an abstraction over UIKit . But SwiftUI can help develop and manage an app in better ways. Also in SwiftUI , we don't have to bother with UI element constraints .That makes the development much more easier .

So we used SwiftUI. When we started , we were told about the existence of different architectures for building mobile apps namely, MVVM, MVC , etc.

Earlier projects were using MVC when we had projects on UIKit.

MVVM is a better architecture that align with the usage of SwiftUI. So we decided to follow it .

We used git for working together as a team and contribute to the project. Here we will see the MVVM Model. The View is basically the UI . The UI interacts ViewModel for any queries for the data displayed on the UI.

The ViewModel Updates the Model whenever its necessary . For example , when we create a task , the View asks the ViewModel to add the task. The ViewModel will take care of making the API Calls and also updating the Model as per the needs.

As you can see the file structure is also organized in similar manner.

This makes the development very easy because each component can be identified by what it does and makes it so much easier for bugfixes.

Also for local data saving, we use CoreData and UserDefaults.

Most of the data are retrieved and passed through API calls . So this was a really good opportunity for us to learn how to do API calls and how its used in real time applications.

1.5 Organization

This report is divided into five modules, and a full description of each module of this project has been provided for the sake of clarity and comprehension. Module 1: This module serves as a formal introduction to the project and contains all of the necessary information. In this

section, we introduce the reader to the project's numerous terms while also outlining the problem or motive for undertaking the project in the first CHAPTER-01: INTRODUCTION Page | 4 place. Along with this, we're also getting started on our project's purpose and the methods that will be used to carry it out. Module 2: This module contains a collection of current research studies that have been conducted in connection with our project. In this section, we place a greater emphasis on the methodology that the articles employed. In addition to this, we're keeping an eye on the results of their various initiatives as well. Module 3: We will go through the many stages of our development in this module, and we will learn about the design and implementation of our approach. In this section, we will also construct the model and attempt to represent it from a variety of perspectives, including analytical, computational, experimental, mathematical, and statistical perspectives, among others. Module 4: In this module, we will examine the performance of our project and provide recommendations for improvement. Module 5: This will be our last module, in which we will discuss the outcome of our research as well as examine our findings and conclusions. In addition to this, we will examine the project's future scope as well as any enhancements that may be feasible in the near future. In addition, we will describe some of the applications in which the system might be beneficial.

CH-02 - INTRODUCTION OF THE COMPANY

About ZopSmart:



ZopSmart a cutting-edge retail technology company that provides all the tools to build e-commerce enterprise Whether you're a traditional retailer looking to establish an omni-channel business or an online-only shop looking to build an e-commerce business, ZopSmart has a suite of tools that will help you get there fast and efficiently.

For a long time, e-commerce has been popular. And we can see that there are several options in the online store market. Assume you're a retailer who wants to open your own internet store. Through the numerous tools available, ZopSmart can assist them in establishing an outstanding platform for their online growth. ZopSmart offers a variety of products that can be utilised by many vendors to launch their own online retail stores. And there are some options that experience can be improved like a self-checkout system, etc.

More than 83 percent of customers feel convenience while shopping is more important now than it was five years ago, according to the NRF 2020 consumer research. This poll was conducted in October of 2019, before we even knew what the term Coronavirus meant. Convenience is no more a "nice to have," but a "must have" characteristic, since shut-downs and social alienation have become the norm. Covid-19 use has risen online, and recent NRF surveys anticipate that demand will continue to grow even after the majority of the population has been vaccinated.

About ZopSmart Customers:

The convenience of shopping online has been addressed by most retailers (at least on the front end). Many people went online before the epidemic, and Covid-19 has forced the surviving retailers to go online as well.

Customers of ZopSmart are primarily from the retail sector.

It covers those who are already involved in the online retail business or who want to get involved.

The Kroger Company, or simply Kroger, is an American retailer that operates supermarkets and multi-department shops across the United States (either directly or through subsidiaries). One of ZopSmart's largest clients is Kroger.

**About ZopSmart Values:**

The value ZopSmart gives to our customers is centred on improving efficiency and productivity. Regardless of the system you use, our goods assist customers in growing as much as possible in the online retail market with the least amount of effort, while also ensuring that they pay a low price, which is achievable because many clients will share the benefits.

With cutting-edge technology and concepts, we will ensure that both the customer and the client are completely delighted.

ZopSmart Vision and Mission:

Our Enterprise platform is used by two of India's largest e-grocers to conduct their online operations. Hundreds of SMEs are using our Eazy platform to expand their e-commerce businesses.

You receive enterprise-class technology at an affordable price when you work with us. Because the cost of development is shared by several clients, you only pay a small portion of the total cost.

Over the course of seven years, we've created and improved nearly all of the capabilities required to run a lucrative e-commerce business. You can get to market in nearly no time with us.

ZopSmart origin and growth:

1. Our retail expertise is based on our team's extensive experience in the fields of FMCG, retail, supply chain, and logistics. All of our knowledge has been incorporated into business procedures and products.

2. Our product has been under development for over seven years and processes thousands of transactions every day. In the realm of retail technology, our portfolio of products is one of the most modern, stable, and scalable options available.

3. ZopSmart is a Best-in-Class eGrocery platform with worldwide reach, with clients in more than 6 countries across North America, the Middle East, and Asia. We

With a modular solution that can interact with your existing ERP and Front End Solutions, you can give end-to-end coverage of your omni-channel journey.

4. ZopSmart's upsell algorithms increased sales by 20% for the retailer.



Products and Functions:

ZopSmart is a Best-in-Class eGrocery platform with worldwide reach, having clients in more than 6 countries across North America, the Middle East, and Asia.

With a modular system that can interact with your existing ERP and Front End Solutions, we give end-to-end coverage of your omni-channel journey.

ZopNow Smart Store:

Let's say a physical retailer wishes to establish an online presence. This is where ZopNow Smart Store comes in. It enables the fastest possible delivery of an online business with the least amount of work.

Features include:

It also assists you in establishing an online and mobile presence using cutting-edge technology.

Smart Product Locator:

A product locator for your store's clients via a mobile app. - Customer location is determined using WiFi or infrared beacons. Provides an exact route to the product. Notifies store employees to assist the customer. If an item isn't found, it notifies store workers. Allows the customer to complete self-checkout.

Smart Payment Gateway

Accept online payments from your preferred payment channels with just one step.

Use an integrated wallet to increase consumer loyalty - Use an integrated wallet to make checkout quick and easy. To encourage repeat purchases, refunds and cashback to wallet are offered. Wallet credit as gift cards for low-cost customer acquisition

Smart POS(Point-of-Sale)

A smart POS system that lets the user and the operator to operate and utilise the system at the same time, with complimentary information displayed on Dual Screens. This includes self-checkout, which is a new concept for a retail store to employ.

Various departments and functions:

- ***TECHNICAL TRACK:***

A Growth, contribution, and leadership are all emphasised on the best technical track, which is centred on a technological core. Although software engineers require people skills to work well in a team, and those abilities should increase over time as their technical skills do, the technical track's concentration should be on technical skills.

What does it take to be a great software developer?

Productivity - Above all, a great software engineer is extremely prolific, frequently producing many times the real commercial value generated by their junior colleagues.

Problem solving — Software engineering excellence cannot be defined without the ability to solve technical problems across disciplines and with systems in various levels of repair.

Quality — An outstanding software engineer's code, documentation, and discussions are continuously of high quality, serving as role models for more junior workers.

Engineers should have a broad understanding of a variety of systems and patterns in software engineering. They utilise this information to improve the design of new systems and to train others.

System design, sometimes known as "architecture," is critical for all software engineers, particularly those who work in organisations that lack specialist architects.

Business/product knowledge - The finest software engineers are deeply familiar with their company's industry and products, and seek to provide the best value for the company, not just for the sake of software.

Data and evidence - Relying on gut instincts or careless debugging are less serious errors. The finest engineers collect just enough data to demonstrate their point and then follow the data to a solution.

Confidence and communication - All of these qualities will be undervalued if the engineer is unable to communicate confidently, clearly, concisely, and inspiringly with both business stakeholders and fellow engineers.

Production support - The greatest engineers understand that their responsibility does not end when their code is deployed. They own their achievements and failures all the way through the manufacturing process.

Planning and execution - A successful project requires systematic and consistent planning from all engineers involved. The best should lead by example in these initiatives.

- ***SALES AND MARKETING:***

Provides excellent customer service in order to increase product sales. Acts as a liaison between manufacturers and customers. To assist sales, makes essential phone calls and meetings. Allows others to ask questions as needed.

Digital Marketing, SaaS Product Marketing, SaaS Growth Marketing (Demand Generation, SEO, SEM, Inbound Marketing, Web Marketing)

Solution Engineering, Value Engineering, Consultative Sales, Digital Transformation Consulting and Advisory are all examples of B2B SaaS sales.

- ***CONSULTING:***

A software consultant assists a corporation in managing and moderating its technology content. Web content (websites and website code), document management, and digital

records and asset management are just a few of the types of software that software consultants may work with. This person helps the firm to handle these components efficiently and safely within the company's guidelines. In a service-based environment, software consultants may also collaborate with clients. In fields like website design and internet marketing, some consultants operate on a client-by-client basis. advertising. They may also work within the organisation, assisting areas such as marketing and social media with software requirements.

Software consultants are normally expected to have three to five years of computer science and consulting experience. A bachelor's degree in computer science is frequently necessary, as is familiarity with and implementation of a wide range of computer scripts and languages, such as Java, HTML, and SQL. Software consultants should also be able to multitask well, operate autonomously, and use sound business judgement.

Solution Architects, Functional Consulting, Project Management, Client Relationship Management, and Customer Success Management are all examples of IT consulting.

CH-03 - LITERATURE SURVEY

DARPA is considering a significant programme aimed at building cognitive technologies that can assist busy government or military personnel in managing and even doing office and military activities. These 'cognitive assistants' will be able to think and learn, as well as be aware of and explain their own behaviour and accept commands from people [8]. A task list manager system (TLM) is one conceivable incarnation of this type of technology that could help users organise and execute their to-do lists. A system like this would:

- Capture the person's daily tasks.
- Plan and execute simple actions.
- Prioritize, manage, and reason about tasks.
- Learn to improve by being told, observing the user, asking questions, and reflection.
- Record notes, action items and ideas.
- Answer questions and offer advice and assist in planning and problem solving.

An early effort was made in our laboratory as part of this project to understand natural task management methods, as well as the types and volumes of work completed by persons who are similar to potential DARPA technology users, particularly busy professionals and managers. We were particularly interested in learning what types of task management demands a TLM could meet.

There are several best sellers and tools on how to organise one's time and prioritise tasks [for example [1, 7]. The market for these resources appears to flourish on people's concerns about prioritising and performing their numerous duties successfully [4]. PIM (organisation, note-taking, reminders, and calendaring) has also been investigated in the HCI literature, but focused mostly on the challenge of organising papers, files, and notes for the purposes of reminding and efficient retrieval, rather than task management [2, 6, 9, 12, 13, 14]. A number of studies have been conducted on how individuals use calendars in practise (for purposes other than event scheduling) [5, 16, 17]. However, this research concentrates on a particular resource that primarily meets time management needs. Furthermore, the various readings on cognition, planning, and task execution available in the traditional psychology literature [10, 15] have little to say about task management and planning in the workplace.

External resources are crucial in accomplishing complex tasks, according to distributed cognition assessments [11, 18], but the research does not look at how external representations serve to help their creators judge the current condition, scope, and priority of many tasks to be performed. Rather than articulation work, the focus has been on resources that assist job execution. (labour required for planning and organising tasks) [19]. And, from an articulation standpoint, task management entails more than just managing physical and virtual collections and scheduling activities. Recent research has begun to address the problem of generic task management in the context of email [3, 20, 23]. Given that many PC users are overwhelmed by the sheer amount of chores they have to complete, This tendency is hardly surprising, as many PC users are overburdened by the sheer quantity of chores handled via email. Any successful task management tool, according to this research, should be tightly connected with email functionalities. We believed we needed to look at task management strategies more generally because email is unlikely to be the complete picture. We briefly describe the results of a "snapshot" pilot study in the remainder of this paper before turning our attention to a longer-term investigation representations and practises in task management We also talk about how we put some of our fieldwork-inspired ideas into an early design. To safeguard the anonymity of our participants, all names of people, projects, and organisations have been changed or concealed in our reporting.

A Snapshot of To-do ists

We ran a pilot study in which three administrative workers, four researchers, and one manager were requested to show all of the resources where they maintained to-dos and count all of the active to-dos they were currently tracking in each one before beginning an in-depth investigation of task management. Based on 595 to-dos counted by the 8 participants, Table 1 illustrates the average number of to-dos and resources employed to represent them. We also recorded explanations for each resource as well as 79 thorough explanations of to-do examples, picking one or two from each resource each individual demonstrated. Some fundamental characteristics of effective to-do lists immediately arose, some of which are addressed below: To-dos are created with minimal effort, hence the majority of them do not define the work; instead, they are often merely detailed enough to serve a useful signal. One task, for example, was to scribble something on a pad of paper: 'Joe the attorney.' "A reminder to send him letter," said the explanation. I believe I was meant to inquire about the [...] litigation with him. "I have no idea." Interestingly, to-do material is frequently

grammatically incorrect, such as "Send Mother's Day" or "Beth blah blah." Because the cue is so simple, it only works for a short time when the job is retained in memory. In certain circumstances, an item such as a book or a stack is sufficient to recall the work without the need for a note. Only a small percentage of to-do reminders are found in lists. We discovered that only 14% of the to-dos we counted were on a list. To-do lists are useful in a variety of situations. They are sometimes included in a list that indicates the quantity of work to be done. They can be consultation resources that link to work objects, or they can be work objects that exhibit state as well as to-do-ness.

Many to-dos are reminders placed in the way in advance of a routine practise that will occur at the appropriate time for the to-do to be noticed. "When I go to get my bag to go home," for example, "I have to take that [item next to bag] home." During email-centered work, emails left in the inbox, in particular, act as prompts.

To-do lists can be displayed at any level of abstraction or detail (see Figure 1). We found one to-do (not displayed) that was detailed enough to permit the production of some slides, while other "deliver presentation" to-dos just mentioned the presentation's subject. To-do lists aren't always completed. People put off doing some things and consider others to be less important. Two people recommended storing low-priority tasks in an electronic format that would be lost if an application or system crashed. A great technique to cut down on your to-do list!

Implications of task-list manager design

The above discoveries had some clear design implications. First, even without the incentive of smart to-do management support, around half of our participants' 70 or so to-dos are already online. Email and the electronic calendar are the primary resources, although both have flaws [3, 20, 5, 17]. Thus, if a TLM can capture this activity, 50% of what is going on may be tractable to improvement, and maybe more if users are inspired to transfer additional to-dos online because of the benefits of system support. However, there are numerous obstacles. A TLM must provide:

- Diverse ways to view and manipulate to-dos to emulate advantages of existing resources, going beyond lists.
- The in-the-way property, e.g., by becoming the habitual place for routine activities where reminders might be encountered.
- Instantly on, to support quick and easy input and clear visualization. PDA's are often abandoned due to slow laborious input and attenuated output [6].

- No formal task description, categorization or decomposition required from users, and any level of abstraction must be allowed for atomic task entries.
- A mechanism for handling stale to-dos of low importance that are diminishingly likely to ever get done but have not been explicitly deleted.

The snapshot study, while useful, left many important questions for a TLM unanswered, for example:

- What help do people need with prioritization?
- What factors drive people to prioritize and complete tasks such that a TLM can propose appropriate action?
- What is the lifespan of a useful to-do?
- What kinds of task management resources are appropriate for different challenges?

A longer-term study was conducted to record and track a large number of to-dos in order to answer these questions. Only 7 people were able to participate due to the lengthy time commitments. Participants were chosen for their ability to multitask and had a variety of work schedules, but none of them used any task-management strategies such as the Franklin Covey system:

- M1: In our organisation, I manage between 5 and 9 research personnel. Examines intellectual property (IP), develops and executes research, and seeks outside funding.
- M2: Oversees our organization's 15 to 20 researchers. Tracks intellectual property and conducts commercial development.
- Prof: University professor and co-director of a research laboratory with a staff of 5 to 20 individuals. Conducts, manages, and secures financing for research.
- SPM: Senior product manager responsible for registering and supervising 200 tour operators who sell their services through his website. His primary office is in London, UK, although he works out of a modest office in the United States.
- DDM: A charity's Director of Development and Marketing. Writes grant proposals, communicates with funders, and manages 3 to 4 employees.
- DM: District Manager for a chain of retail food and beverage retailers with ten locations. Visits locations, manages a team of 20 store managers, and tracks and builds business.
- SAM: Sales account manager for a large retailer and wholesaler of office products. There are around 300 active and potential accounts.

The following was the data gathering procedure (used at all of the participants' workplaces):

- A two-hour preliminary background information interview
- Four 1-hour weekly task-tracking interviews (referred to as TT1, TT2, TT3, and TT4, respectively) in which (usually) 10 to-dos that could be completed by the following week were elicited, discussed, and rated for importance and urgency on a scale of 1 to 5 were elicited, discussed, and rated for importance and urgency on a scale of 1 to 5. Every week's to-do list was reviewed during the next interview. Participants included prompted not to focus on a single to-do resource or only important and urgent to-dos.
- One day of shadowing, during which the subject was watched performing routine duties.
- One-hour final interview to answer any outstanding questions and check on the status of all to-dos.
- All of the sessions were recorded on video and transcribed.

For 37 codes such as 'done,' 'on-hold,' 'common,' and 'discretionary,' all 287 to-dos were coded with a value of 'yes' or 'no.' Cross-coder reliability was found to be 92 percent for a subset of 50 tasks (50x37=1850 codings). Pearson's R and Spearman Correlation tests were used to find correlations between coded components.

CH-04 - TOOLS AND TECHNOLOGIES

Swift

Swift is the culmination of cutting-edge programming language research and decades of experience developing Apple platforms. Named arguments have a simple syntax that makes Swift APIs considerably easier to learn and manage. Even better, you won't have to type any semi-colons. Modules reduce headers and offer namespaces, while inferred types make code clearer and less prone to errors. Strings are Unicode-correct and employ a UTF-8-based encoding to enhance efficiency for a wide range of use cases, so they can handle foreign languages and emoji. Memory is automatically managed via strict, deterministic reference counting, which keeps memory use to a bare minimum without the burden of garbage collection.

Xcode

XCode is an integrated development environment (IDE) for Mac OS X. It includes a set of Apple-developed software development tools. We can use XCode to create software for macOS, tvOS, iOS, and watchOS. The most recent stable release of XCode is 11.0, which is available for all macOS Mojave users on the Mac App Store. We'll go through numerous XCode contexts in this segment of the training.

SwiftUi

SwiftUI is the new method to build apps for iOS, iPadOS, watchOS, and macOS.

It's a major departure from the "traditional" way of doing things, obliterating several current Apple frameworks including UIKit, AppKit, and WatchKit.

All of these frameworks have one thing in common: they are all necessary.

Pixel by pixel, you, the coder, decide how things should look. Then you react to user actions and manually update the data. You determine how the UI should update with each modification.

SwiftUI is a complete overhaul since it is reactive, and the user interface reflects the current state of the data. There will be no more "linking objects" as in UIKit.

You also write far less code. If you've ever created an iPhone app with UIKit, you'll always wonder, "Is that it?"

When it comes to programming, SwiftUI is all about writing code. StoryBoard and Interface Builder are no longer available.

This is ideal for me since I can keep my code in Git and observe the changes made over time without having to wade through XML nonsense.

If you've never dealt with UIKit before, you won't understand what I'm talking about. Don't worry, that's excellent for you.

We have so many benefits since Apple was able to start from scratch with SwiftUI.

Github

GitHub allows numerous developers to work on the same project at the same time, which lowers the chance of duplicate or conflicting work and speeds up production. Developers may use GitHub to concurrently write code, track changes, and come up with new solutions to problems that may occur during the site development process. It may also be used by non-developers to create, modify, and update website content.

Charles

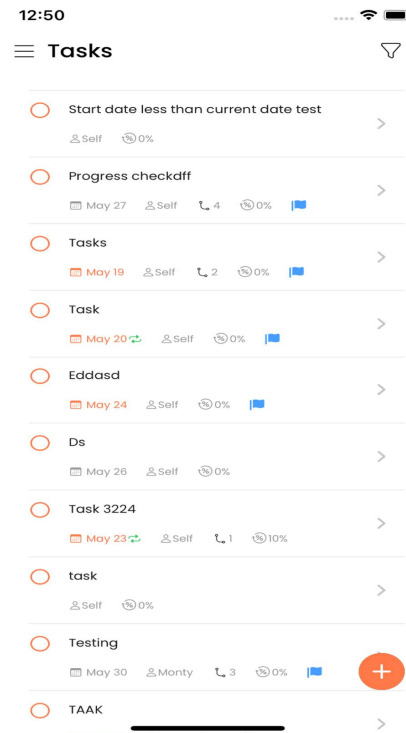
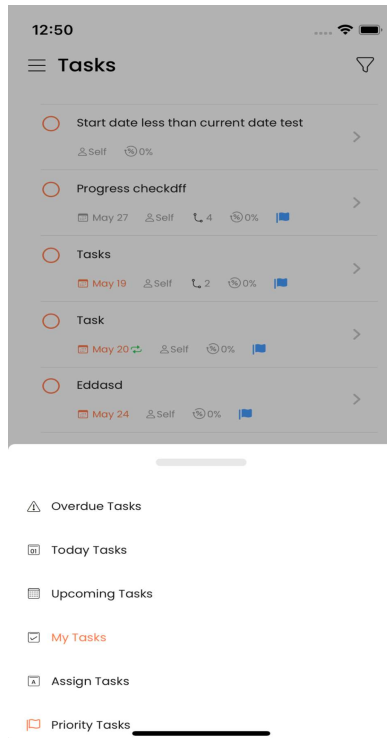
Charles is a web proxy that runs on your PC (HTTP Proxy / HTTP Monitor). Your web browser (or any other Internet programme) is then set to use Charles to access the Internet, and Charles is able to record and display all data transmitted and received for you.

You can't see what's being delivered and received between your web browser / client and the server in web and internet development. It is difficult and time-consuming to pinpoint the specific location of the defect without this visibility. Charles makes it simple to observe what's going on, allowing you to rapidly identify and resolve issues.

Debugging with Charles is rapid, reliable, and advanced, saving you time and hassle!

CH-05 - SYSTEM DEVELOPMENT

Application Design



This is the main homeScreen View of the tasc-ios application, where we can see in the top bar view , it contains two buttons. One at the left corner that is for going to the ham menu view which contains different columns like my Teams, Settings, Notifications history and Profile View.

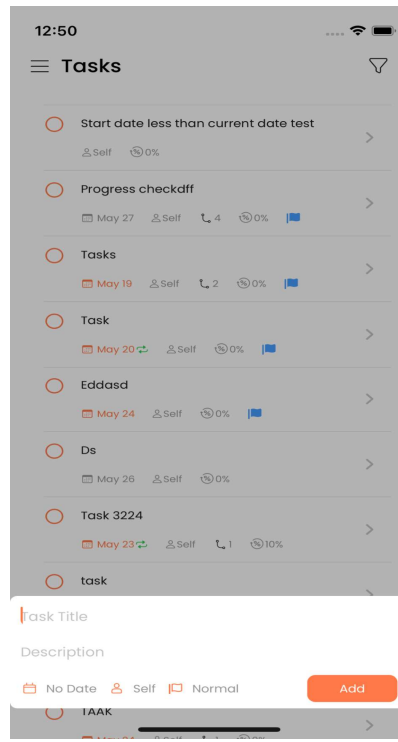
This screen contains list of tasks providing which filter is applied, the default filter for the homescreen is myTasks.

The task card contains lot of information you can see, the first one is the dueDate of the task, if the user doesn't select the due date, the date will not appear in the task card.

The second one is the assignee name , i.e to whom that task is being assigned, the third one is the no of subtasks, and simultaneously followed by progress of that task and priority indication by different color flags.

This screen design has a vertical stack followed by list of Hstack i.e of taskCard view

Inside the task card we have use the VStack , the first element of that VStack is the title of that task and the second element of that VStack is the HStack which contains the following information described above like dueDate, no. of subtasks etc. On the left side is the bottom sheet of the filter view which contains the different filters, when we tap on the any filter, the tasks are sorted accordingly on the homescreen view and the botttomsheet gets disappeared.

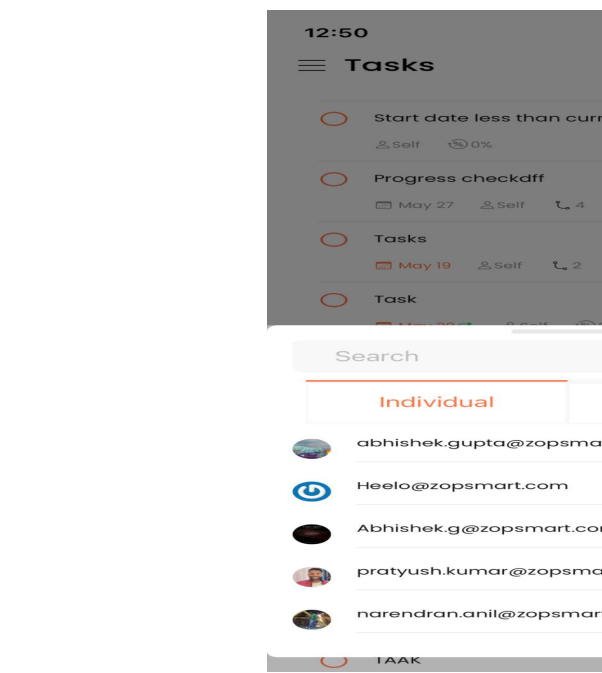


This comes the bottomsheets for adding the task in the app , You can see there are different information that we can add while adding the task i.e it's title, description dueDate, assignee name and it's priority.

When we click on any of them a different bottomsheets will open up and we can fill or select the details from that bottomsheets .

The title is the mandatory field while adding the task , if we do no add the title while adding the task , A warning will appear which will indicate the user that title is the mandatory field.

For implementing the bottomsheets a special method is used by making different view and calling that view through a boolean variable inside the view where we want that bottomsheets to appear like homescreen in this case.



This bottomsheets will appear when we want to select the assignee for the task, as you can see in the view, a segment view is implemented which has two columns Individual and Teams. When we tap on the teams, a list of team names will appear and we can select any particular team and when we click on the individual in the segment view, a list of individuals will be displayed and we can assign that task to any particular individual, also a search bar is implemented through text field and a functionality is added as when the user type something in the textfield, it gets searched in the list depending on the segment view selected i.e individual or Teams.

Further is the popUp view for selecting the priority of the task, below is the view for that. when we tap on the priority, a popup view will appear indicating the the different priorities via different color flags.

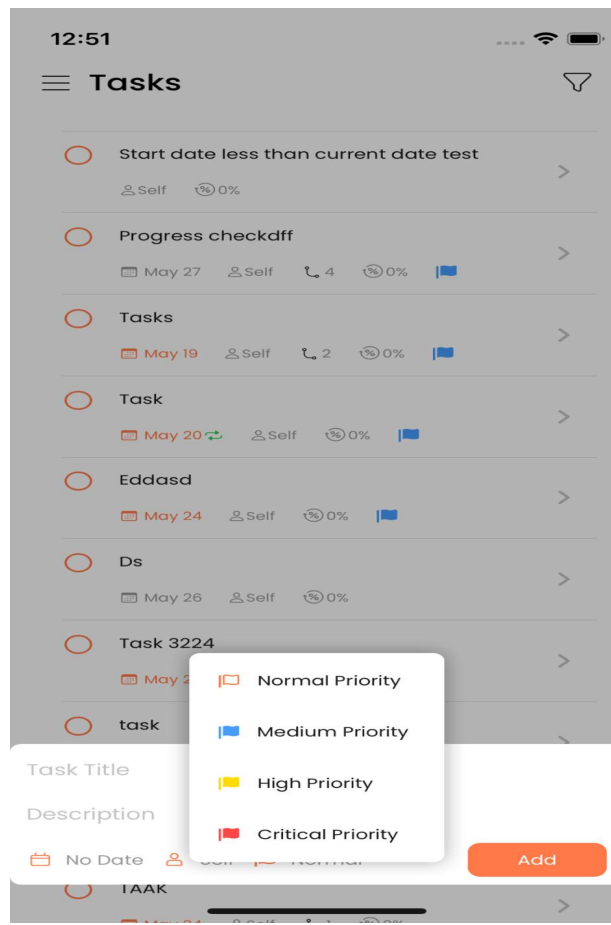
Normal flag is indicated through flag with empty body and orange color

Medium priority is indicated through blue flag

Critical priority is indicated through red flag

And the last high priority is indicated through yellow flag.

Below is the popup view for selecting the priority.



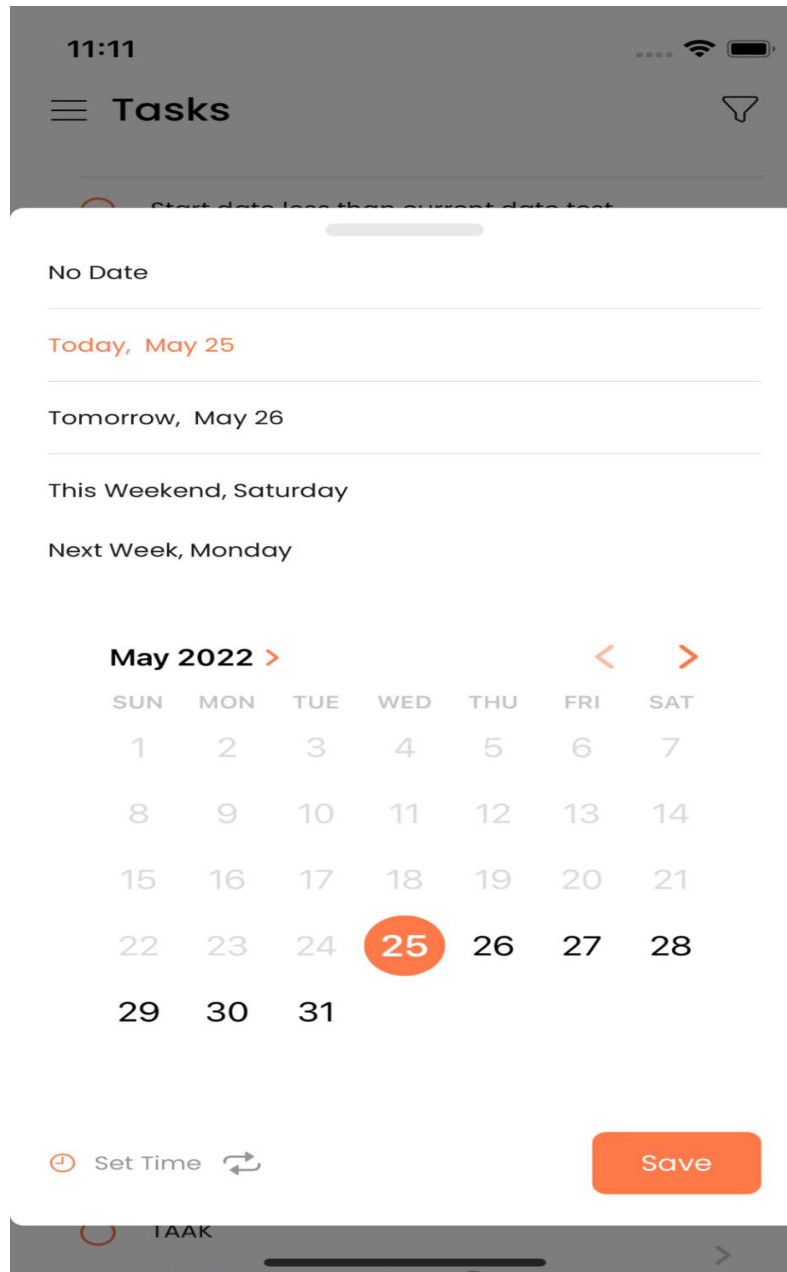
Now comes the date and time view, below is the bottomsheet implemented of that, there you can see the repeat button at the bottom of the sheet and a timepicker button at the left corner.

When we click on the right button, a different view mainly a different bottomsheet will appear indicating the four different rows called daily, weekly, monthly and yearly and when you click on any on of them, a different bottomsheet according to the selected repeat mode will appear and you can select the details accordingly like you can select the week day or you can select the start and end date or you can select the particular days in the month depending on the repeat mode you have selected.

A timepicker view is there, the swift default library is used in the default timepicker, when u click on the button, apple default timepicker will appear and you can select

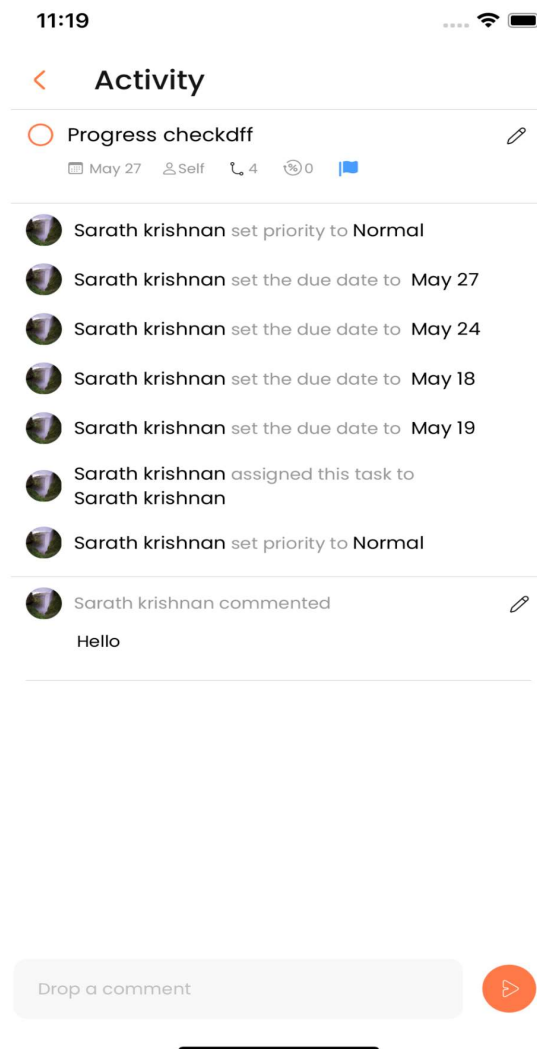
the time accordingly.

If the user doesn't select the time, the current day midnight time will be taken i.e 11:55 PM



The default date selected as you can see is the current date and user cannot select the date less than the current date and after selecting the date, time and whether the user want that task to be repeating or not. The user can simply click on the save button and that details get saved.

Below is the screen for task activity where you can see the history of that task.



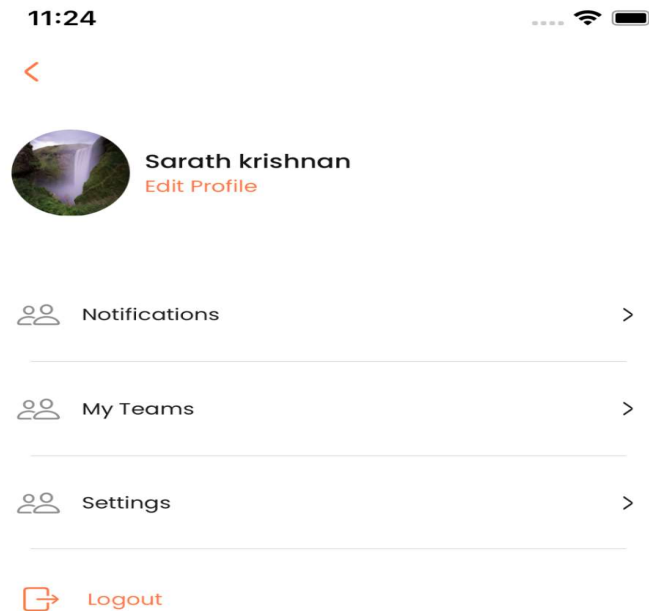
We can see in the bottom, the comment option where user can type the comment and the comment will be displayed in the task activity screen

User has the option to edit that comment anytime and real time will be displayed at the top that when the user has comment on that task.

User has the option to edit the task from the task activity screen, you can see the pencil at the top of the screen where user can press and bottom sheet will appear with prefilled details of

that task and after editing the details, user can simply press the edit button and that task gets edited.

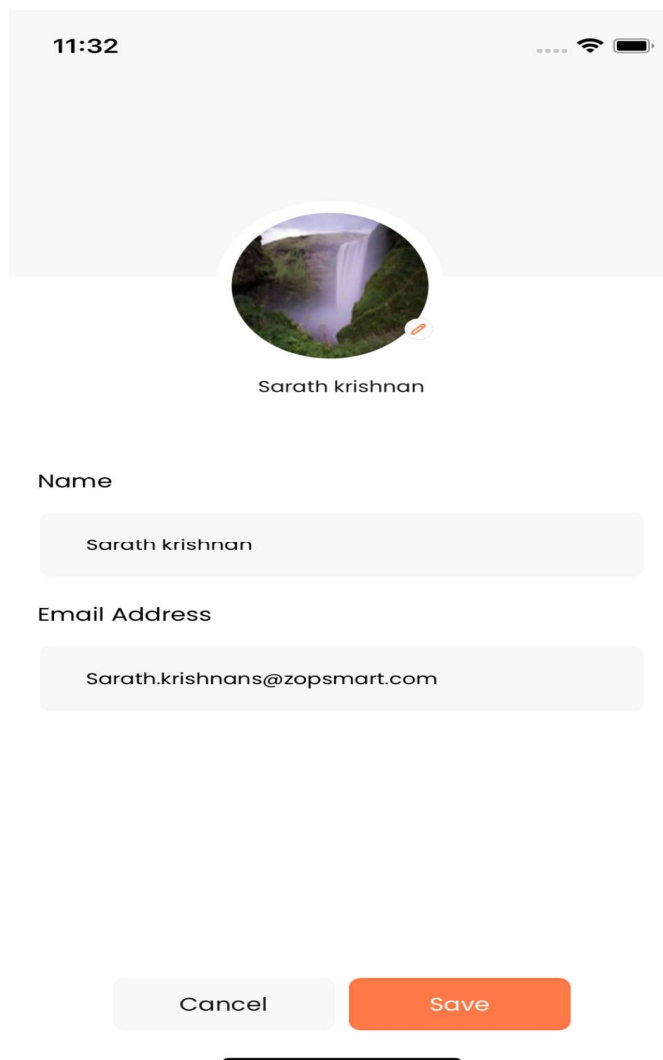
One more feature is there, you can see the circle at the top left, it the button basically , when the user taps on that button . that task status gets completed and task will be gets disappear from the screen.



Here you can see the hammenu view , which is made of basically the two different views namely userProfile view and profile view , these two views are placed inside VStack and that VStack is embedded inside the ZStack

The upper view is called the userProfile view and the bottom part is the profile view which contains different columns which navigate you to different screens like notifications history, my teams, settings and the last one is the logout button and when we click on the button, the a

popup will appear which asks the user to logout and user has the option to press YES or NO .
If the user press the YES button, the user will be redirected to login view and if the user presses the NO option, the popup will disappear and the user will stay in the same view i.e hammenu View.



A mobile application popup window showing a user profile. At the top, the status bar displays the time 11:32, signal strength, Wi-Fi, and battery icons. Below the status bar is a circular profile picture of a waterfall. Underneath the picture is the name "Sarath krishnan". The form contains two input fields: "Name" with the value "Sarath krishnan" and "Email Address" with the value "Sarath.krishnans@zopsmart.com". At the bottom, there are two buttons: "Cancel" (light gray) and "Save" (orange). A black horizontal line is visible at the very bottom of the screen, likely representing the home indicator bar.

11:32

Sarath krishnan

Name

Sarath krishnan

Email Address

Sarath.krishnans@zopsmart.com

Cancel Save

This is the edit profile view , you can see the user can edit his or her details anytime. Photopicker is used for selecting the image from the system gallery and for editing the name and the email id, the textfields are used. Once the user has selected the photo and the name and after clicking on the SAVE button, a loader will come and when the Api call gets over , the loader will disappear and the user is redirected to hammenu view.

< Notification

New Tasc for you! Task 'sl' has been assigned to you	May 23
New Tasc for you! Task 'task' has been assigned to you	May 19
New Tasc for you! Task 'notifi' has been assigned to you	May 19
Task has been updated! Heeeeeehh	May 19
Task has been updated! Incomplete	May 19
Task has been updated! Fdfdfdd	May 19
Task has been updated! We're	May 19
New Tasc for you! Task 'hiiooo' has been assigned to you	May 17
New Tasc for you! Task 'hiiooo' has been assigned to you	May 17
Task has been closed! Heyyyyy	May 11
New Tasc for you! Task 'Heyyyyy' has been assigned to you	May 09
New Tasc for you! Task 'Heyyyyy' has been assigned to you	May 09

This is the notification history screen, in this you can see that the use of VStack is made which contains the list of HStacks.

An Api call is being made in this screen where the user will get all the notifications history along with the date at the right side of the notifications card . Spacer() is used for making this distance apart i.e one of left corner and the other thing on the right corner .

When we click on any of the notifications, the user will be redirected to the task activity screen of that particular notification same as when we tap on the particular task in the homescreen.

< Settings

- Assignment** ∨
 Select the kind of assignments you get about your activities.
- Task is assigned to me
- Task is assigned to team
-
- Task Closed** ∨
 Select the kind of tasks that are beigng closed.
- A task created by me gets closed
- A task where i participated gets closed
- A task in my team gets closed
-
- Task Update** ∨
 Select the kind of task updates that you get.
- A task created by me gets Updated
- A task where i participated gets updated
- A task in my team gets updated

This is the settings view, here you can see that the settings are basically for receiving the notifications, if the user has enabled the particular notifications, then only that user will get that notification. This view is made with the help of toggle() method i.e. when the user presses on the small arrow, then only that small view will open to its fullest otherwise not. API call is being made when we click on any particular checkbox and get API call is being made when the user opens the screen.


11:45





< My Teams


Search





-  Zopmart (11) >

-  Task (2) >

-  Task - iOS (5) >

-  monty (14) >

-  team tasc (2) >

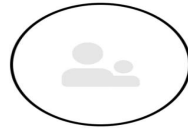
-  Huu (4) >

Add New Team

11:45



< Team Details



Team Name

New team name

Team Owner

Sarath krishnan

Cancel

Save

Here you can see the two different views that is the My teams view and the add new team view.

The design of the view is quite understandable , at the top of the add new team is the image selection where user can select the image from the gallery

The next is the team name and the owner of the team which by default is the user only.

After pressing the save button, the api call is being made and user is redirected to the my teams page with the new team added on it.

If the API call fails the warning message will appear indication that an error occurred after redirecting to the my teams page.

The first photo is the my teams view where you can see that the first is the HStack and inside that the hstack is the textfield where user can search from the list of teams already present.

ForEach loop is used to get the list of teams , a default swiftui list is used for displaying the teams name .

Both the views contains one back button which is totally customized according to the app, this back button has an action called self. Presentationmode. Wrappedvalue. dismiss()

This is the functionality for dismiss the current view and redirect the user to the previous screen.

The default button in the swiftui is the orange button with same arrow and a text name BACK is written on it.

In the team card view as you can see in the first photo, there is the number in the bracket which indicates the number of members that team contains.

If the team is empty and doesn't contain any member, by default it will show 1 number in bracket as because owner will always be the part of the team and if the owner also leaves the team, the team will get deleted.












The arrow thing which you are seeing in the team card view is because of the navigation link we have applied in the list view to redirect user to particular team page showing all the members of that team.

12:30



< Zopmart



-  **Narendran**
narendran.anil@zopsmart.com Owner
-  **rid**
ridhdish.desai@zopsmart.com
-  **Shruti Khare**
shruti.khare@zopsmart.com
-  **Monty**
Abhishek.g@zopsmart.com
-  **Ruppal**
Ruppal@zopsmart.com
-  **Rakshit**
rakshit.singh@zopsmart.com
-  **Vikash Kumar**
vikash@zopsmart.com
-  **Vijayl**
vijay.sethiya@zopsmart.com
-  **Sarath krishnan**
Sarath.krishnans@zopsmart.com
-  **zopsmarttester@gmail.com**
zopsmarttester@gmail.com
-  **Rakshit Singh**
rakshit.lko@gmail.com



This is the particular team view as you can see the list of members in the team and if you are the owner of that team, a button called add member will show in the bottom and when that button is pressed, different view will appear and from that view user can select the members he or she wants to add from the list or by simply typing the email id in the search bar and when the email id is written fully, the add button will show up and by clicking that add button, that member gets added to the particular team and the user will get redirected to the team details view.

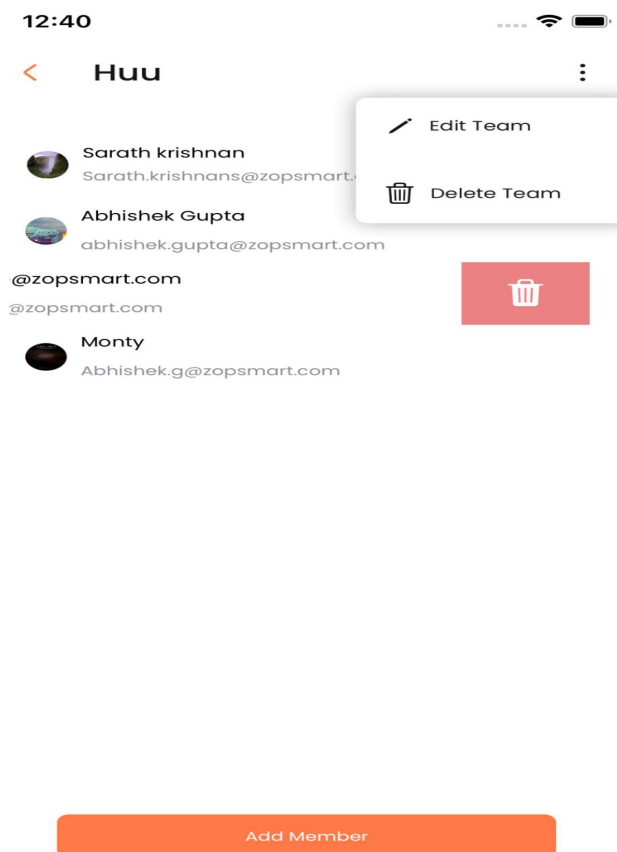
If the user is the owner of that team, the user can simply swipe from the right and the user can simply remove any member from the team.

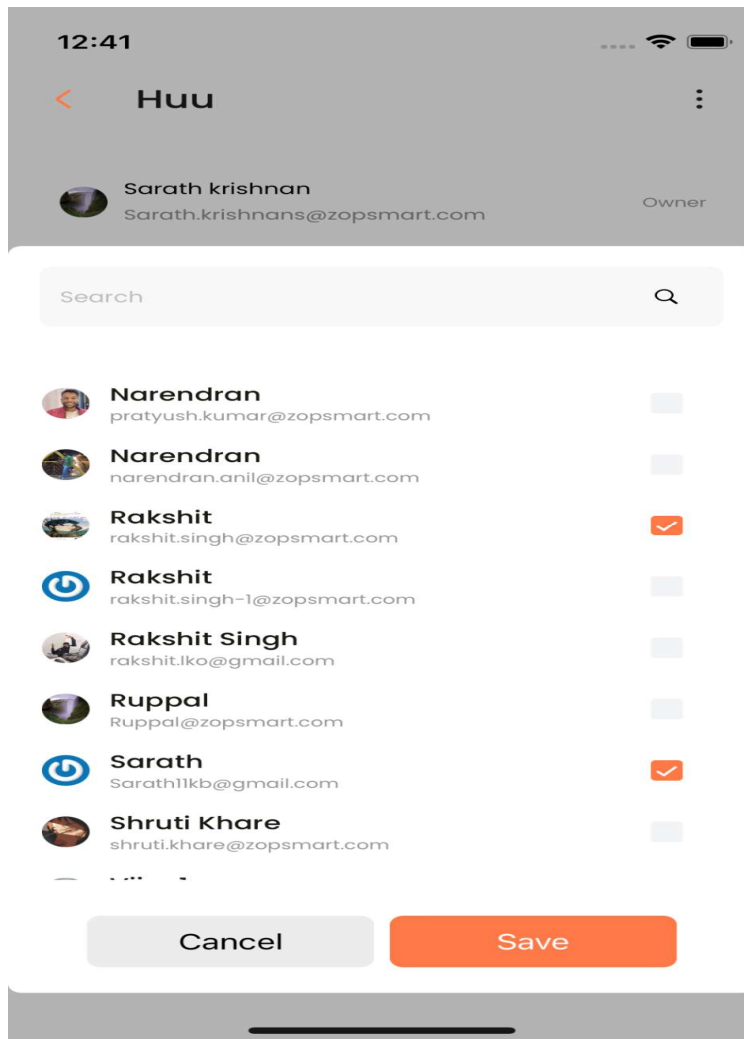
After deleting the member, the snackbar will appear which has an undo button on it , and if the undo button is pressed, the member will again gets added and if the undo button is not pressed within 5 seconds, the snackbar just gets disappeared.

Many more features are also there depending on if the user is the owner or not .

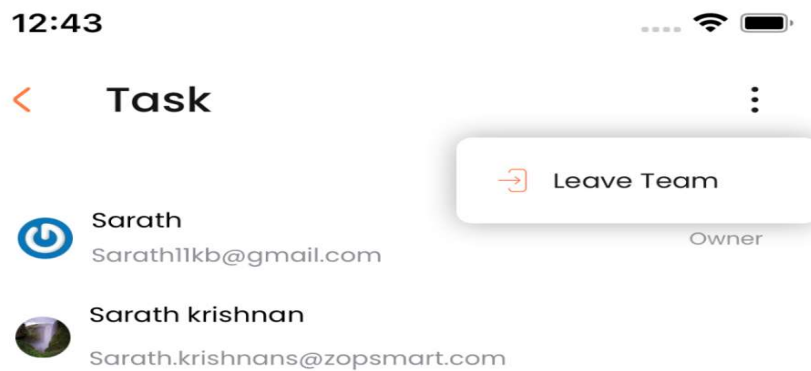
If the user is the owner, the user will see the two options i.e basically of edit the team details and the other is for deleting the particular team and if the user is not the owner of that team, the user will see only one option i.e of leaving that team.

All these screenshots are added below.





This is the view for the add members view and you can see the two different methods i.e from the searchbar and the other one is from the list of members that are already part of some team.



Next comes the Auth Views which contains different views like signin, signup, forgotpassword and change password.

If the new user is signing into the tasc app, the user has to signup in the app by filling all the details like name, email-id etc.

Next comes that if the user has forgot the password, the user can reset the password using forgot password, once the user clicks on the forgot password button, the user will receive the email having the default password and from there the user can login into the account.

Next comes the main functionality i.e the change password, the user can change password anytime by filling the old password and setting the new password according to his or her own choice.

All the views are pasted below, the theme for the app is orange color which can be customized anytime.

1:05



Login to Tasc

Enter your details below

EMAIL*

PASSWORD*

[Create new account?](#)

[Forgot Password?](#)

1:06



Login to Tasc

Enter your details below

EMAIL*

Already have an account?

[Sign In](#)

1:06



Create an Account

Enter your details below

NAME*

EMAIL*

PASSWORD*

CH-06 - CONCLUSION

Conclusion

The application made is basically the todo list app which helps in the management of different tasks. Allow your users to swiftly add new activities to Application using our very user friendly method by utilising our Quick Add endpoint. You may assist your customers find material within your app faster and return more frequently by automatically attaching links to your app to the task content. Quick Add is the quickest way to get started using the Api which are made using GoLang technology.

We can add different information along with the title and description, the main thing we can add the due date of the task, next we can add the priority of the task i.e how important is that task. Along with this we can assign it to some individual or particular team.

The dueDate has an important characteristic i.e we get a notification when the due time of the task remaining is 5 minutes. And also there is an Api made for the notification i.e when any user get assigned any task or his team get assigned any task, the user will get the notifications. Several features are there in this app like the login setup is proper there, the old user can simply log in to the application and the new user can make the new account.

Also there is also support for the forgot password, when the user forgets his or her credentials, he or she can reset the password by simply sending the request to particular email id , once user clicks on reset password, after entering the email-id, the user will get redirected to the login page and default password is sent to that email-id. Along with this the change password functionality is also there, i.e user can create the password of his/her choice by using the current password, with only one condition that the password should be atleast 6 characters long.

The most important feature of the application are repeating task, i.e the task can be made repeating i.e on daily, weekly, monthly, and yearly basis. The user can select the start, end date and particular days of week or particular date, when the user wants that particular task to be made repeating.

Next any user can comment on task, the comment of that task is visible in the task activity screen, once commented, the user can edit that comment.

Also we can edit the task which are created at any time, the changes made to the task from its level of being created to current time, all the editing history is visible in task activity screen.

After tapping on any particular task, user is redirected to task activity screen.

This task has an important characteristic of having as much subtasks, there is no limit of number of subtasks of any particular task, all the subtask has one field called parentId, which is same for all the subtasks which is equal to the taskId of the parent task.

Every subtask can have weight or not, depending on user adds the weight to that subtask, as much number of subtasks are completed, there total weight is calculated and that number is equal to that parent task progress.

This subtask has all the same features and fields as one having task with only one condition i.e all the subtask has same parentId which is equal to the taskId of the task.

We have the concept of teams i.e user can create the team and he can add the members to the team by only knowing their email-id, the default owner of the team is the user itself, the owner has the right to edit and delete the team and members has the right to leave the team. Various snackbars are added at the appropriate places, i.e after deleting the task, subtask, deleting the team and removing the member from the team of which only the owner has the right.

A snackbar is visible for 5 seconds, with undo button on it, if the undo button is pressed, that action will be reversed, if the user doesn't press that undo button, the snackbar will just simply dismiss after 5 seconds via completing that action.

Image upload feature is also there in the user profile and team image, we can upload the image of our choice, if any particular hasn't uploaded his or her image or the team doesn't has any image, the app will take image from the default gravatar instead of not showing any image.

Notifications column is there in which the user gets all the list of notifications till now, the notifications which the user has seen had different colour and the notifications which are not seen by the user till now had a different background color.

A filter view is there in the homescreen, which contains different features like priority tasks, overdue tasks, upcoming tasks, my tasks etc. When the user clicks on any particular filter, the homescreen is loaded with all the tasks after applying that particular filter.

A settings column is made which contains different checksums depending upon receiving the notifications like receiving the team notifications, individual notifications etc.

The application name is tasc which is available on the AppStore for android and the authorization for ios is under progress, once the ios app is approved by Apple, IOS app will be live on AppStore.

We find it challenging to keep focused on our tasks and complete our work efficiently when we are distracted by social media and other conveniently accessible online diversions.

Furthermore, moving between tasks frequently may give us the misleading impression that we are productive when we are not. Rather than focusing on erasing tiny items from our todo list only for the sake of looks, we should prioritise chores and work on the most critical ones.

This app's objective is to assist us become more conscious of how we spend time when performing things and how productive that time is.

It can help us put some limits on social media to prevent distraction and keep track of how much time we spend working on our to-do list. We can better organise our daily routines when we have a better understanding of the expected time we'll need to spend on our duties, as well as the validated time spent on the things for reference or personal/team reviews.

Basically in today's busy world, it is very difficult to manage the daily task. This app is basically the solution to it. We can customize it according to the user need, along with different features and functionalities. The dueDate and time plays an important functionality which notifies the user when the date and time is get to over.

Also this application also indicates the task which are completed and which are pending.

Future Scope

This application has the innovative future scope, like how many of us can finish the daily tasks on time, so we need a everytime planner which can help us to plan and manage our tasks accordingly.

In this project, we can add a daily tracker statistics graph, where you can see the overall progress and can see how your graph is going ...up or down

In this busy world, we need an everytime reminder so we can complete our daily tasks on time and in coming time, this app would be a great resource for that which has the high futuristic scope.

Mentor's Review

He performed admirably during training; he is a quick learner who takes less time to comprehend concepts. At the same time, he has exceptional problem-solving skills, making him an invaluable asset to our organization.

Pratyush Kumar
Team Lead

REFERENCES

1. About Zopsmart Technologies from ZopSmart.com <https://zopsmart.com/>
2. Swift Official Documentation <https://www.swift.org/>
3. McAfee official website <https://www.mcafee.com/>
4. Kroger website <https://www.kroger.com/>