

発表資料

・目次	1分	1
・導入(開発背景など)	5分	5
・feseekの機能説明	15分	15
・チーム全体で苦労したこと	10分	5-7
・個人の発表 1人2-3分 (18-27分)		20
・まとめ	3分	3
・謝辞		

合計 約50分

○メモ

個人発表の好きなイベントはおまけ要素
時間稼ぎと個性を出す場として設けたので、
役職→担当したこと→苦労したこと→今後の課題→好きなイベントの順で
自由に2～3分発表
すごくゆっくり話してください

どっちがどこを話すのか役割分担も決める

リハーサルは月曜日の朝から発表時間まで

個人と全体で分けてやる
何時から何のリハーサルをするとか具体的な時間を決めておく

PPTを読むのは×
文章を書かずにキーワードで

導入

今回我々が研修内で習ったこと→springbootとreactについて

皆さん、地域のイベント、どうやって探していますか？

・今回の課題は大きく分けて2者(利用者、主催者)がいることが条件

↓

- ・美容系介護系グルメ系などが出た中、イベント口コミに。
- ・過疎化で人が少なくて盛り上がらない。
- ・そもそも存在を知らない/知られてない。

- ・少子化、過疎化が拡大しているから地域活性化できるようなものを。

feseekの機能説明

ゆっくり実際に動かしながら説明する

- ・地域イベントのロコミサイト

チーム全体で苦労したこと

- ・共通認識
- ・意見をまとめる
- ・人数多くて大変
 - ・全員に意見を聞くと時間かかる
 - ・リアクションが難しい
- ・SpringBootとReactつなげる

皆に一人一個聞く

個人の発表 1人2-3分 700文字ぐらいを個人で考える

- ・役職
- ・やったこと
- ・個人的に苦労したこと
- ・今後の課題
- ・好きなお祭り一つ？

https://www.canva.com/ja_jp/presentations/templates/pitch-deck/

●レビュー

アプリ名の説明

二者間とは？どういう二者間で作ったのかの説明があるといい

背景の前に開発の話

開発→開発背景

実際のところに予定を色を変えて一画面で見せる

目標達成状況の理由

なんでその%なのか

誰視点なのかの説明を画面に入れる

間に合わなかったはいらない

最後にまとめていれた

できましたじゃなくてこういう機能がありますと言い切る

ゲストとは？

管理者の画面ある？

できなかったけどなんでできなかったのか

時間が足りなかっただけ？

何があればできたのか？技術が足りなかったのか？

チーム全体で苦労したことは一項目一画面にしたほうが文章を読まれない

個人 中途半端な改行を修正する

まとめ 伝えたいワードを2つぐらいドーンと乗せるぐらいの方がいい

一般利用者と主催者がいるという紹介

アプリ名の意味の紹介

原稿

これから祭りだわっしょいの発表を始めます。

●目次 畑中

こちらが今回のアジェンダになります。まず導入として7月の研修で学んだこと、開発目標、開発背景についてお話しし、開発したものの機能説明を行います。そして、チーム全体で苦労したことに

についてお話しして、個人の発表を1人2、3分ほどさせていただいて、最後にまとめという流れで進めさせていただきます。

●導入 小嶋

7月の研修で私たちが習ったのはSpringBootとReactです。

SpringBootはJavaでWebアプリケーションを作る際の定番のフレームワークとしてバックエンドのSpringBoot

それからReactは近年人気のある、こんかいのフロントエンドとなっています。

また、今回の開発するwebアプリの条件として二者間でのやりとりを扱うものを作ることになりました。この点を踏まえたうえで私たちが開発するアプリを考えました。

ここでスケジュールのこともいれる(ガントチャートを表示)

続いて開発における目標とスケジュールについてです。

今回の目標は全体的な目標として共通認識を持って開発に望むこととなっています。

それに加えてコーディングに力を入れるという観点から全員が成長することができることを目標に設定しました。

続いてスケジュールについてです。7月の最初の週にSpringBootとReactの講義を受け、開発期間は大体三週間になります。

こちらが開発当初予定していたスケジュールと実際のスケジュールになります。

成長重視の観点もあり、コーディングをなるべく重点におく計画となっていますが、前回までの想定利用者と異なる2者間であったことから、その二者間のどちらにどのような機能が必要になるか、またどのように利用者の権限を分けるのかなどの内容から話し合う時間がかかり押していました。

下に出ている紫のマーカーが実際の我々のスケジュールになります。先ほど申し上げたことからコーディングに入るのは予定より1.5日ほど遅れてしまう結果となり、なかなか予定通りには事が運びませんでした。

今回の目標としてメインである共通認識を持って開発に望むことは特に開発当初は意見が出せていなかったことや、各ファイルなどで連携不足が出ていた。しかし、最後に向かうにつれてよくなったことから70%ほどの達成度ではないかと考えられました。

次に成長重視に関しては各々が1週間での短いコーディングのなかで努力が見られたが、完成までは至らなかったもので80%ほどが適正かと考えます。

次に今回我々が開発したアプリの内容に入っていきます。

突然ですが皆さんは小学校や中学校に通っていたとき、地元のお祭りには参加されましたか？地方にはたくさんのイベントがありますね。みなさんはそのイベントをどこで知りましたか？親や友達にきいた？それとも掲示板に張り出されていた？自分自身に身近な地域のイベントがあったとき、その情報を伝達しているほとんどは地域内であると思います。

そんな中昨今では少子化や過疎化の問題が進んでおり、それに追い打ちをかけるように新型コロナウイルスが流行しました。これらによってイベント自体が減ってしまっています。

現在ではようやくこの騒動が落ち着いてきたゆえに外出することを前向きにとらえる方が増えてきていると思います。だからこそイベントに参加したい人、人をたくさんよんでイベントを盛り上げたい人が多くいるように思えます。

そんなとき地域のイベントの情報を伝達しているのが地域内の存在であるということがそれぞれの意図の障害になってしまいます。その障害をなくし、イベントの主催者とイベントに参加したい一般のお客さんの二者それぞれを結びつけられることを目標に作ったのが今回私たちが開発した『feseek』になります。

●feseekの機能説明 畑中

ここからは今回開発したアプリ「feseek」の機能説明をさせていただきます。

まず、こちらがトップページのイベント一覧ページになります。現在はゲスト画面となっていて、上部メニューのログイン・新規登録はコチラをクリックしてログインすると、上部メニューや機能が切り替わるようになっています。→

ログインページに遷移しますと、下のリンクから新規登録ページに遷移することができ、→
一番下にあるプルダウンから一般利用者か主催者かを選択することができます。→

ログインページに戻りまして、ログインの際IDパスワードを間違えて入力してしまった時は、エラーメッセージが表示されるようになっていて、忘れた際もログインボタンの下にあるリンクからパスワードを再登録できるようになっています。→

こちらがパスワードの再登録ページで、新しく登録したいパスワードを二回入力して再登録することができます。→

続いて、こちらが一般利用者のアカウントでログインした状態のイベント一覧ページです。ゲスト画面の時と上部メニューの項目が変わっているのがわかると思います。ログイン・新規登録はこちらの項目がログアウトに変わっています。そして先程は無かったプロフィールの項目が増えています。→

一般利用者ができることとしては、一つ目に、イベントを検索する機能です。イベントをキーワードで検索することができます。イベント一覧ページにある検索ボタンをクリックするとモーダルウィンドウが開き、キーワードを入力して検索するとモーダルウィンドウが閉じて、キーワードにマッチしたイベントが一覧で表示されます。→

そのイベントの一つをクリックするとイベント詳細ページに遷移し、イベントの詳細が見れるようになっています。イベント名から開催日時、アクセスなどの基本情報がまとめて見ることができます。→

下の方に行くと、ロコミとタイムラインが表示されていて、ここでログインしている一般利用者のみ、ロコミ投稿ボタンが表示されるようになっていて、ボタンをクリックするとモーダルウィンドウが開いて投稿することができます。右側にあるタイムラインは変更された情報やイベント日の状況などを主催者が投稿することで、それを一般利用者は閲覧することができます。こちらは主催者のアカウントでログインした時のみ、タイムライン投稿ボタンが表示されるようになっています。→
そして一般利用者は上部メニューのプロフィールからプロフィール情報を確認することができ、さらにプロフィールの編集、アプリの退会、そして過去に自分自身が投稿したロコミを一覧で見ることができます。→

こちらがプロフィールの編集ページで、変更できる項目はパスワード、ユーザー名、メールアドレスの3項目になります。→

続いて主催者のアカウントでログインした際のプロフィールページです。一般利用者の画面とは違い、右下のボタンが主催者自身が投稿したイベント一覧になっています。→

そしてこちらが投稿したイベントを確認することができるイベント一覧ページになります。こちらでイベントの編集と削除、イベントの投稿もここで行うことができます。右側にある画像がイベント投稿のモーダルウィンドウの画面になります。→

どちらのプロフィールページも、退会ボタンをクリックすると確認ダイアログボックスが表示され、間違えて退会してしまうことを防ぎます。→

主催者の一般利用者と同じ点は、上部メニューと検索機能で、両者とも同じように利用できるようになっています。また、ログアウトの時もどちらも確認ダイアログボックスが表示されるようになっています。→

以上がfeseekの機能説明になりますが、今回、時間の余裕がなく、実装できなかったものがいくつかありますので、そちらも紹介させていただきます。

まず、お問い合わせ機能です。こちらは一般利用者がアプリの利用方法の確認や悪質な口コミなどを発見した際に通報、お問い合わせできる機能として実装する予定でした。けれど、現段階での技術力では実装が難しく時間の余裕もなかったため断念した機能となります。ですので当初は管理者画面も実装予定でしたがお問い合わせ機能がなくなったため、管理者画面の実装もなくなりました。

また、ログイン時のID・パスワードの再設定に関して、当初はIDも再設定できる予定でしたが、こちらは今回時間の都合で実装できませんでした。

そして、口コミで画像を投稿・表示する機能、検索で詳細検索ができる機能、表示だけとなってしまった新規登録機能とパスワード再設定機能、口コミとタイムライン機能に関しましても、今回は時間が足りなかったため実装できなかった機能になります。また、お問い合わせ機能を除いた機能に関しましては時間に余裕があったら実装することができたのではないかと考えております。

●チーム全体で苦労したこと 小嶋

続いてチーム全体で苦労したことについてお話いたします。

こちらはチームの「全体」ということもあり複数人におけるコミュニケーション不足からくるものが多かったと見受けられます。

まず一点目として、フロントエンドであるReactとバックエンドであるSpringBoot、それからデータベースにわかれて作業をしたため、作っておいた1ファイルにどのような内容を記述するのか、それから表記揺れのような、都度コミュニケーションをとらなければなにかが違ってしまいうような内容が多いことが挙げられます。解決策としてお昼休憩後と17:40過ぎに行った進捗確認でそれぞれの進捗状況を報告しあいました。特にファイルの大きな変更点などは後から見てもわかるように議事録に見やすいように記述しておくなどの工夫をとりました。

二点目は、人数が多くなった点です。人数が多くなったということであればふつうは分担ができ、一人当たりの作業が分散するため速く作業が終わるように思えます。しかし一人が5日かかる作業を5人で行えば1日で終わるということがないように、チームメンバー内での認識のずれや、作業がかみ合わないということが起こってしまいます。これらで特に注意すべき点といえば、ツールとしてGitHubを使っていることになります。つまり複数人が同じファイルを扱うことによりGitHub上で衝突を起こしてしまう可能性が存在することになります。実際に衝突してしまうようなアクシデントが何度かありました。これに関してはメモ帳に予備のデータを記述しておいたり、プッシュする前に何かデータがないかを確認したりと対策をしていました。もちろんGitHub以外の観点でも全体で苦労した一点目と同じようなコミュニケーション不足からくるものが多かったです。それに加えて人数が多く2者間を対象とした機能が必要になるということで、内容となる機能も多くなり、言い方が少し悪いですが、妥協点といえるような部分を見つけられなかったことが完成まで至らなかった可能性を含んでいると考えられます。

三点目として一日という比較的長い間いなかった人への情報伝達に苦戦しました。1か月という長い期間であることから、どうしても全員がそろわないようなタイミングがあるものです。今回もそれがありませんでしたし、今後我々が仕事をしていくとしたら必ずと言っていいほど誰かがいないという場面に遭遇することになると予想できます。このようなときに議事録として文章をまとめていないと見える形として情報を伝えられなくなってしまいます。アプリを開発途中であるならば画面やコードを。画面自体を設計しているならばその作成途中のものを見せればそれでいいというわけにもいきません。話し合いの結果に至るまでもヒントが隠されているからこそそのようなことが

あったのかどうしてこうなったのか、いなかったなりに伝わるようなニュアンスや表現で情報を伝えなくてはならないことを学びました。

四点目として各自の細かい表現方法の違いがあります。これは6月の発表の時点で経験した人が多かったものがまた起こってしまっているように思います。名前の差異や、使う関数の違いが影響してしまいエラーが出てしまうことになります。わかりやすい例としてスネークケースとキャメルケースのような明確なちがいを持たせることで名前の違いが出づらくなると考えられます。ただ、コードの記入においては講義内で習った教材が共通のファイルであっても個性が出るものであるものですし、たとえばある人は講師からいただいた資料を参考に、別の人はインターネットの記事を参考に、これらがそれぞれでエラーはでなくとも名前や関数から紐づけができないでエラーになることもある。そういったことも人が多いから6月よりも多かったように思えます。

最後にリモートであることによる問題です。インターネットにつながったPCがあれば作業ができる。こういった観点はエンジニアという仕事についての強みです。しかし、いざ対面で仕事をしていない状況ではどうしてもzoomのようなツールを頼りながらコミュニケーションをとることになります。そうなると対面しているとみえるような細かいしぐさなどは見えなくなります。実際に話していると、これは些細なことのようにあっても相手が反応を示していることが話している側も理解できます。これがないからこそとりまとめる方も意見を募りにくく、そうでない人たちは喋りだしづらいという欠点を持ちます。特に開発が始まった当初はこの影響が強く、意見が出ない上に進行が遅れてしまうというよくない状況になってしまっていました。

チーム全体としての課題は以上となります。
つづいて各個人の成果と課題にうつらせていただきます。

●まとめ 小嶋

まとめに移らせていただきます。

今回の7月の研修ではSpringBootとReactについて教わりました。

これらは様々な機能を内包したフレームワークであり、直接の記述やインポートをすることで様々な機能を一気に扱えるものでした。しかし開発途中ではSpringBootの方でサーバーエラーが起こってしまったり、Reactの表示が完成されておらず実際の動作が見れないなどのこちらだけで解決できないことが多々ありました。これらは一方だけで完成するものではないと思わせると同時に、フロントエンドとバックエンド両方に詳しくなることでどうすべきか判断ができると思わされました。

今回は開発の時間もあり、それらは役割で分けることはしかたないにせよ、役割で別れてしまったことでコミュニケーションが不足してしまう側面もありました。

ただ、これらも開発の終わりが近づくにつれ、zoom内で移動が多くなったり、slackで内容をきいたり成長が見られました。

今回の開発内容として2種類のユーザーに対する機能を考える中で、そもそもどういったアプリをつくるのか、どのような機能を持たせるのか。実際に導入されているアプリが複雑であることを改めて実感させるような事柄が多かったです。

機能自体の取捨選択やどう見せていくか。こういった点でもっと工夫や簡略化ができたのではないかと考えられます。

●謝辞 畑中

最後に謝辞としまして、今回、約4ヶ月の研修でお世話になった全ての皆様、参加させてくださった企業の皆様に心より御礼申し上げます。→

以上で祭りだわっしょいの成果発表を終わります。
ご清聴ありがとうございました。