# 6月4日

チーム名：DIV （ディブ）

由来：イタリア語(divisi)が由来。音楽用語で1つのパートが2つ以上に分かれるという意味から派生して、多様的に物事を考えられるチームという理由。

divタグのように「１つのまとまり」という意味から、一致団結するという意味も。

1時間の目標

・方向性を定める（対象者・概要、目的）~14:00

・ペルソナ作成 ~16:00

・ペルソナを基に、意見を選択 ~17:00

//・選択した意見を細分化 ~17:00

本日の目標

・ペルソナのアイデアを出す（明日は詳細を詰めるだけ）

ゲーム性重視

・キーボードダービー

概要：ランダムで各キーが何度入力されたかをカウント。

　入力回数を予想し、競う？

当たったら景品的な何かを贈呈。

・あすけん

概要：食べたものを入力。そのカロリーを予想し合う。

　お昼ご飯ダービー的な。

・ミニゲーム集

概要：数当てゲームやなぞなぞがいっぱい遊べる。

　一定時間パソコンを操作していると、突如ミニゲームが開始。

　クリア後、「休憩しろよな！」などのメッセージを表示

ストレス解消・癒し

・育成ゲームアプリ

概要：ログインすると何かが育つ。

　　育成段階に応じてポイント付与。アプリ内アイテムの購入のみ可能

・ほぼ日手帳

概要：自分が記入したら相手の記入したものも見られる。

便利機能

・時間管理アプリ

概要：タイマー係の役割をコンピュータが担う。

　前の作業が押した場合、後ろの予定も遅らせることができる。

・Github

概要：今だれが何をしているかを確認できる。

進行中のタスクを記入することで、社員の現在の状態を把握できる。

現在の様子をより詳細に。このような仕事をしています。

Teamsの現在のステータス状況みたいな。スケジュール共有。

テンプレートを作成するフォームも。

質問する相手に通知。

育成アプリorポジティブ発言などの要素も。

目的：用件を伝える際、手が空いているかを確認する手間を省く。

　→手の空いている人を確認。質問可能な方を探し、行きやすい環境を作る

融合案

・タイマーとミニゲーム集

　概要：タイマー機能でプロジェクトを進行。

一定時間たつごとにミニゲームを発生する。

・ダービー系アプリ

　概要：お昼ご飯や運動時間をユーザーに入力してもらう。

　その入力値を予想し合う。

　意見：入力過程が面倒に思う方も。

　 　→コンピュータがデータを集計。人間は予想のみを行う。

・ふぐ山さんを掘る　〇

・微修正後、ふぐ山ステータスを作る（欠点を見出す）

　後輩との関係があまりよくない。

　→一定時間たつごとにミニゲームを発生によって、会話のネタに。

・アプリベースでペルソナ作成

# 

# 6月5日

・本日のコース目標

ペルソナに寄り添った要件定義書を作成する。

チームメンバー(杤本さん)を知ろう。

午前中はもう一人のペルソナを考える

午後から要件定義

→11:30から要件定義開始に変更

アプリの案→マスオさん

・予定や現在の状況を共有できるアプリ

・匿名投稿サービス

・匿名目安箱サービス

・社内Q&A共有ツール(質問や動画)

・匿名で質問や回答ができるアプリ

・定時で帰ると何かが育つ(退勤時間管理)

・魚に特化したレビューサイト

・釣った魚で図鑑を埋めていくゲーム(退勤時間管理と組み合わせる)

アプリに何かを育てる機能を入れる

育つものは上限を決める

アプリの案→明子さん

・食事管理アプリ(社食の購入履歴)

・プロフィールを見れる掲示板

・スキー仲間募集

・アウトドアのおすすめの場所共有

・犬の散歩コース共有

・日々の自慢投稿

・目安箱サービス(悪口などは投稿できない)

・格言

まとめ？

・何か育てるアプリ

→ペルソナからアイディアを出した中で、これなら使いたい！と思えるシステム（アプリ）

匿名目安箱や質問ができるアプリ使いたい

伝えづらい不満なども投稿できる

意見を上司だけでなく社内全体の人が見れる

社内で使えるもの

過去の質問やよくある質問は回答を用意しておいて閲覧できる。検索機能をつける。

機能3つ以上

・育成（例えばログイン1ｐ、質問1ｐ、ベストアンサー1ｐ、定時退社1ｐ）、経験値とポイントでかせげる？

→育ち方：

①ポイントを装備などに交換して、キャラクターに付与（可能ならこっち。着せ替えアプリ？）

②ポイントで育つ

→カンストした場合：

①ならカンストという概念がなくなる

②「今まで育ったキャラクター」欄に図鑑のように表示される？

装備はサブ機能としてもいいかも（やりこみ要素）

機能3つ以上

〇育成（ログイン1ｐ、解答1ｐ、定時退社1ｐ）

〇匿名質問

〇質問・回答の検索

〇目安箱

・格言の表示（日替わり）

デザイン

・育てた何かが後ろで歩いてる（常時）

・おしゃシンプル

会社が定額で購入することとする

wikiのようなページほしい

人によっては見られたくない情報なども載せられるように

文章だけでなくファイルなども投稿できるように

勤務時間外は機能を制限する→勤務時間外はキャラクターが寝ている？（スマホに実装した場合）

質問具体例

・有休の取り方

・オフィスカジュアルってどこまで？

→マニュアル系が多め？

質問内容に関するルールは作る？

→仕事に関する質問のみ

会社に関する要望を投稿するページ（目安箱）

今日の格言募集ページ

作るページ

　・ホーム

　・質問一覧

　・質問詳細・回答

　・育成

　・キャラクター到達度一覧ページ

　・目安箱

　→送るだけ？どこ宛なのかは選択制

　→設定としては目安箱の一覧を見れる人は企業による選択制 (管理職のみ閲覧可能など)

回答意欲をキープするには？

→育成機能

→一定数のランクに到達すると商品と交換可能に

→熱中しちゃう勢への対策として上限決める

社用アドレスで登録

キャラクターデザイン：社員全員が同じキャラクター１体を育てる。本ツールをうまく使うことによって、進化していく。社員キャラクター一覧ページに自分が育てたキャラクターまで表示、それ以外はシークレット表示。その進化段階にいる社員数のみ表示される。

今後の検討案（発表会で言えそう）

・一定数のランクに到達すると商品と交換可能に

→育成目的と離れてしまう

なぜ実装しようと思ったか

→回答意欲の維持

・ポイントを装備などに交換して、キャラクターに付与

のぞいた理由：開発期間が間に合わない

・キャラクター１体が育つという設定は永続性に欠けるのでは？

→話し合いでは、「キャラクターを数種類作成して、その各キャラクターに進化段階がある」という案も出た。今回は期間的に却下。

アプリの名前候補 　順次追加

1.「ちゃらんぽらん」×

2.「アノニくん」￥¥

3.「とくめぇ～」(ヤギ) ￥￥￥

4. 「DIVelop」

キーワード：匿名、質問、育成、目安箱

キャラクターの名前候補　順次追加

1. 「アノニくん」→とくめぇ～のアノニくんが妥当？

2. 「ちゃらんぽらん」

3. 「」

明日の取り組む課題

　・要件定義書

　・アプリ名・キャラクター名

　・外部設計（Webページを可視化→PowerPoint？）

# 6月6日

＜コース目標＞

・今日中に要件定義書を完成させる。

・メンバーついて知ろう（中田さん）

午前中の予定：外部設計の講義

要件定義書作成

＜将来的展望＞

管理者側が目安箱等の意見から個人を特定したい時には、メールアドレスなどの個人を特定できる最小限の情報から

＜リーダーより＞

・全員でもう一度名刺管理アプリを共有（今日中）

・要件定義書はスラックへ（講師のフィードバックを待たずに次へ）

・機能設計等で躓いたら、要件定義書の開発の背景

・後に話し合う内容

　・警告文の表示場所について

　　フォーム送信（機能）

デザイン案

・目安箱　表示方法

　・共感数の多い順に表示

　・目安箱の内容をホーム画面に表示する（格言の代わりに）。展望

・メニューバーに常に検索機能がついて回るのではなく、各一覧画面上部に検索機能を実装

実装時利用できそうなシステム（名刺管理アプリ）

・上に戻るボタンの固定

・セレクトボックス

# 6月7日

コース目標

・チームメンバー（藤井さん）を知る。

・画面設計を完成させて機能設計作成にはいる 。

～村上さんから～

・外部設計→機能面文章化

・目指すレベルの共通認識化

・チーム内の課題を細分化

・画面設計はボタンと項目で十分

作るページ

　・ログイン（初期表示）OK

　・ホームOK

　・質問一覧（検索も）OK

　・質問詳細・回答　OK

　・質問フォーム　OK

　・目安箱一覧　OK

　・目安箱フォーム　OK

　・育成（自分のキャラクターのページ）OK

　・キャラクター到達度一覧ページ　OK

　・ユーザー登録フォーム（メニューバーの表示はログイン後に管理者のみに表示されるイメージ）OK

　計10ページ

＜今後も検討しなきゃかもな部分＞

・ホーム画面のキャラクターはptの説明を話している風の画像を（吹き出しつける等）

・メニューバーの項目の表示の仕方

・検索は質問に対して全件検索

・（質問詳細）ヘッダー、メニューバー、回答送信フォームは固定

　→代替案：質問、回答の部分を別でスクロール可能に

・メニューバー：アノニくんの到達段階

　→文言は今後いいものがあれば変更

# 6月10日

＜コース目標＞

・DB定義書を完成させる。

・メンバーについて知ろう（山口さん）

＜連絡事項＞

・経験者と未経験者を確認

・発表の方向性は今週+来週で決める

　どんなふうにまとめるか　作っていった過程に基づいて発表

・内部設計書の前に入出力の関係性を明確にする

＜レビュー＞

・パスワードの文字列：ハッシュ化するなら64

・メールアドレスと合わせた方が整合性が高い

　→今回なら100,100済

・質問データは同じでもいい（案）

　→見せたいものは質問の内容ではなくさらに読み込む機能

・一覧　全検索はエラー？

　→一覧でいいのでは？

　〇→一覧は目安箱にもあるからそろえる意味で

　　　→質問検索は追加機能

＜見つかった課題（今後の展望）＞

出来たら・・・

・質問の削除機能：質問者のみ

・文書の更新機能：質問者のみ

村上講師に質問することリスト（済）

・質問と回答は別テーブル？

・段階別のユーザー数集計はDB？それとも他のやり方？

　→SQLは単純な方が良い

　　一年の期間内のsumを算出　実行結果をJavaでリストデータにする　jspかサーブレットで表示

募集中と解決済みの切り替えはラジオボタン

募集中・解決済み・すべての質問

＜追加機能＞

検索で募集中か解決済みかをフィルタリングする機能

# 6月11日

＜コース目標＞

　・進める意識を持って作業に取り組む。

　・積極的にチームメンバーに相談する。

　Servletに（getmethod, postmethod）の部分

↓↓↓↓↓↓↓↓↓↓↓ あってる？ ↓↓↓↓↓↓↓↓↓↓↓

// ログインページにフォワードする

RequestDispatcher　dispatcher

= request.getRequestDispatcher("/WEB-INF/jsp/login.jsp");

dispatcher.forward(request, response);

の部分を作る

＜本日の成果＞

　・各ファイルの役割分担（一部）

　・Jsp作成

　・modelの作成（完成）

　・DBのsql作成（着手）

　・アイコン・イラスト案作成（着手）

＜リモートでわからなくなったら＞

・「いいですか～」と声かける

→答えられる人はマイクON（可能な限り!）

・返答なし or 解決が難しいとき

→議事録にメモ（後に話し合い、村上講師に質問）

・余裕があれば、何が分からないかを整理してから聞く

・「何が分からないかわからない」グループ内でわからない点を考える

# 6月12日

＜本日の流れ（始まり）＞

・みんなで1つのDAOに着手

↓

・ペアに分かれて残りのDAO作成

＜コース目標＞

・DAOを終わらせる

・チームメンバー全員が理解できるプログラムを作成する

＜連絡事項＞

担当を割り振ってから作業する

何か見ながら作れる程度でいい

進捗共有する

質問は質問チャンネルからする

<保留>

<佐竹講師にもらったDate型に変換するコード>

if (card.getTorokubi() != null && !card.getTorokubi().isEmpty()) {

Date sqlDate = Date.valueOf(card.getTorokubi()); // yyyy-MM-dd形式の文字列をjava.sql.Dateに変換

pStmt.setDate(10, sqlDate);

}

else {

Date sqlDate = Date.valueOf("2000-01-01"); // 未設定の場合でも何かしらの日付が必要

pStmt.setDate(10, sqlDate);

}

ログイン時に今の時間を

　→sql「update テーブル名 set カラム名 = current\_timestamp」で対応。

　Date型は日付のみ。時間が欲しかったらDateTime型

LoginServlet isLoginOKの戻り値がnull→ログイン失敗

　がnot null→ログイン成功

＜つまづきポイント＞

　LoginServletのエラー

session.setAttribute("id", new Users(1, "メールアドレス", "パスワード", 100, "2024-06-12 14:38:00"));

　改善後

session.setAttribute("id", new Users(1, "メールアドレス", "パスワード", 100, new Date(10000000000L)));

　希望（以下ができれば改善後の方法で実装可能）

・取得した日時データをlong型で取得する。

・取得した日時データを年、月、日に分けて取得する。