

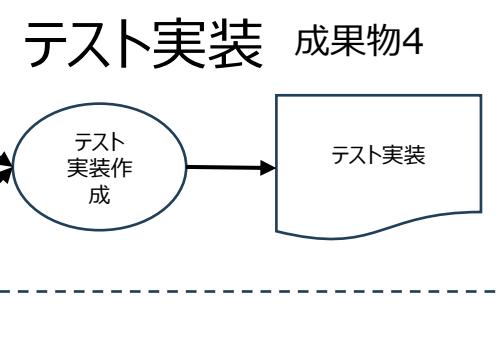
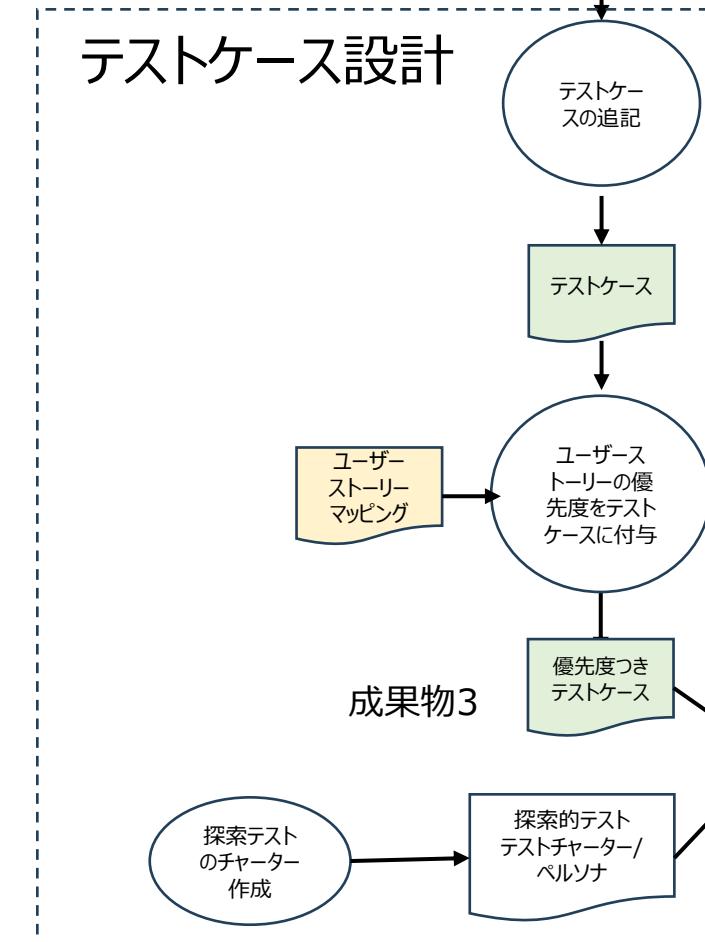
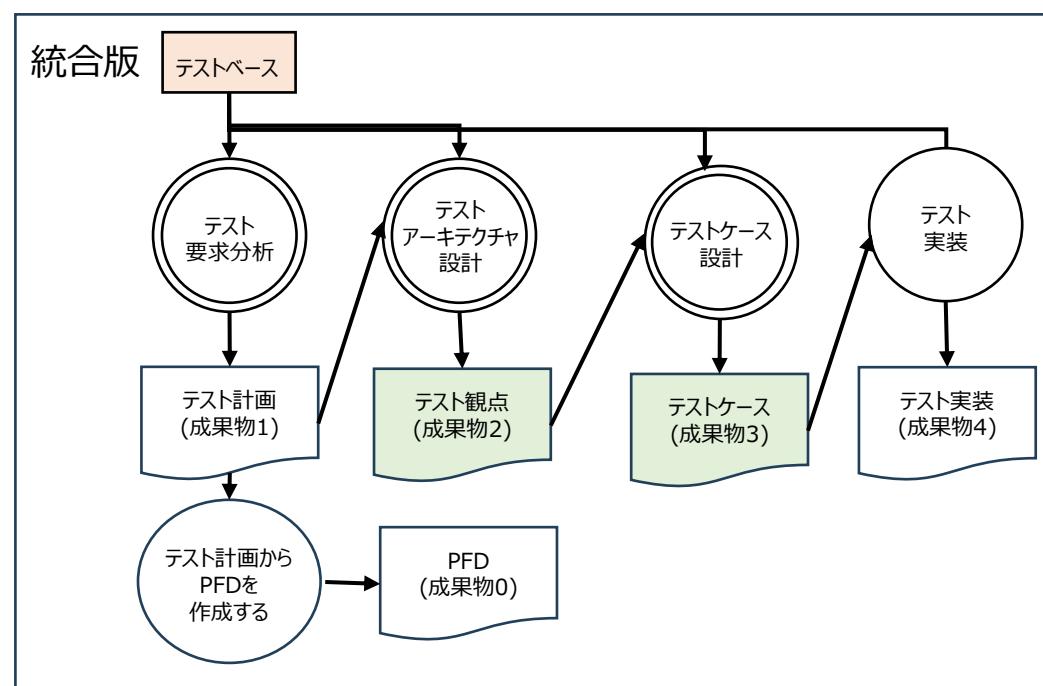
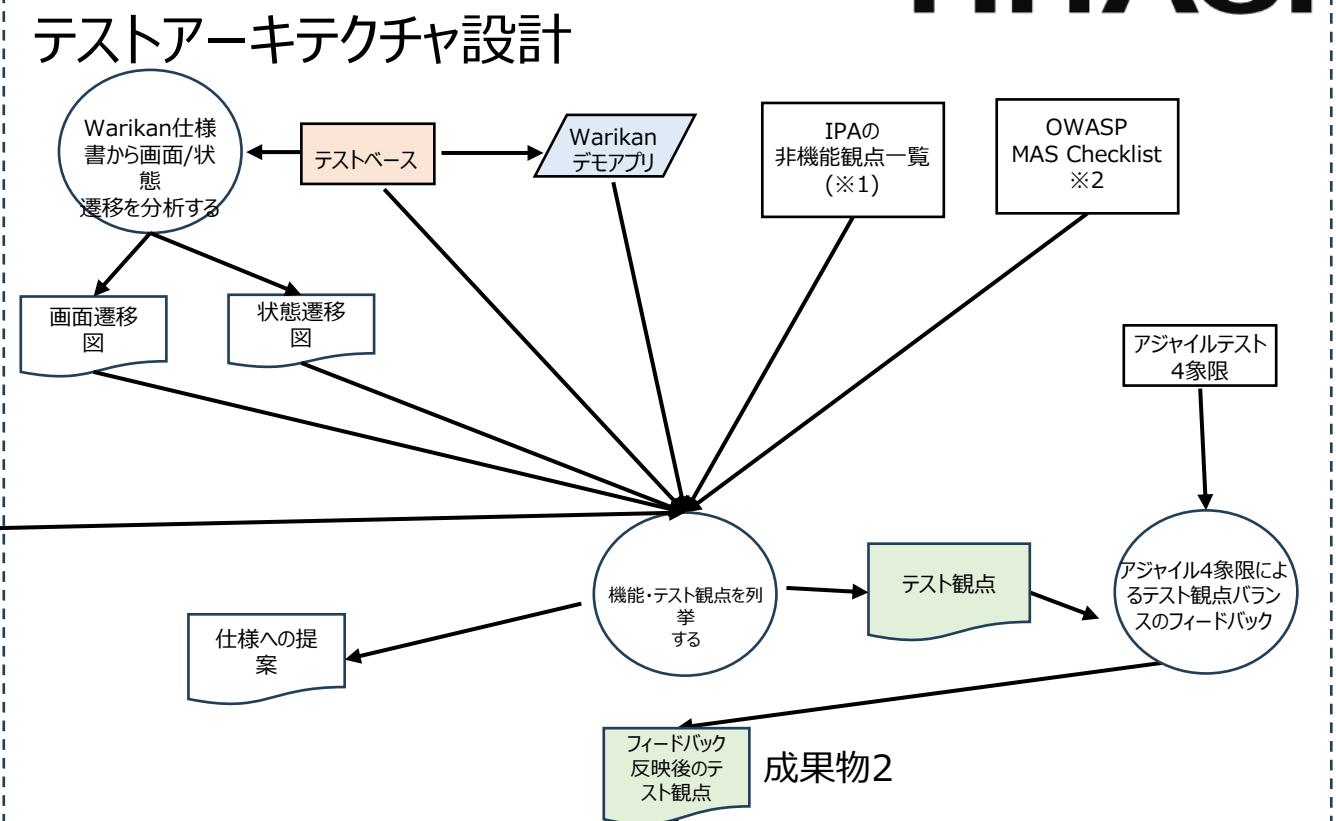
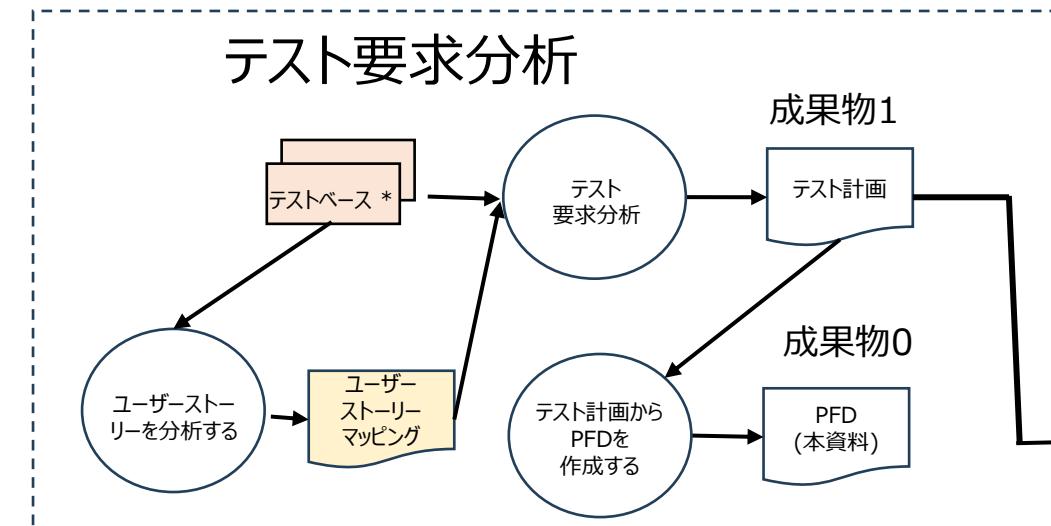
## テストコンテスト出場の背景

- ・ 個人の経験内容をwebアプリケーション分野へ踏襲することで今までに習得した知識をより汎用化したい
- ・ 同じテストベースに対するテスト設計を共同で進めることで、互いの経験内容の違いの認識と新しい考えの蓄積を図る

## コンセプト

### チャレンジングなテスト設計

- 互いの経験を踏まえて意見を出し合い、互いの"やってみたい"を形づくりたい
- チャレンジした部分について、参加者の方から意見を頂き自分のものにしていきたい



(※1) <https://www.ipa.go.jp/archive/digital/iot-en-ci/jyouryu/hikinou/ent03-b.html>  
 (※2) [OWASP MAS Checklist - OWASP Mobile Application Security](#)

## ユーザストーリーマッピングにてユーザのユースケースを洗い出し PFDにてテスト設計全体像を可視化

### ユーザストーリーマッピング(抜粋)

テスト対象の機能について、ユーザー目線の価値で整理

ス	ユーザー	ステータス
1	サービスを利用したいユーザーとして、新規登録ボタンからブラウザの会員登録ページを開いてアカウント登録をしたい。なぜならアプリを使う前に会員情報を登録しておく必要があるからだ。	nice_to_have
2	サービスを利用したいユーザーとして、IDとパスワードを入力して認証に成功したらログイン画面から次の画面へ遷移できるようにしたい。なぜなら自分専用のデータにアクセスできるようになりたいからだ。	nice_to_have
3	サービスを利用したいユーザーとして、パスワード入力中は文字が●で表示されるようにしたい。なぜなら周囲の人に内容を見られずに入力桁数だけ確認したいからだ。	nice_to_have
4	サービスを利用したいユーザーとして、ユーザーIDとパスワードに文字数や文字種のバリデーションをかけてほしい。なぜなら入力ミスや弱いパスワードを防ぎ、安全にログインしたいからだ。	future
5	サービスを利用したいユーザーとして、認証に失敗したときにエラーメッセージで原因を通知してほしい。なぜなら何を直せばログインできるのかすぐに理解したいからだ。	nice_to_have
6	サービスを利用したいユーザーとして、ログイン画面のチェックボックスでIDやパスワードを端末に保存できるようにしたい。なぜなら毎回同じ情報を入力する手間を減らしたいからだ。	future
7	サービスを利用したいユーザーとして、保存用チェックボックスを外したときに入力欄を未入力状態に初期化できるようにしたい。なぜなら共有端末などで自分の情報を残さずログアウトしたいからだ。	future
8	飲み会の幹事として、自分側と相手側の人数や金額をバリデーション付きで入力し、スライダーで支払い割合を調整できるようにしたい。なぜなら正しい値で割り勘計算を行い、公平な負担にしたいからだ。	mvp
9	飲み会の幹事として、人数と金額の入力欄に「1~99」「1~999999」のような範囲ガイドを初期表示してほしい。なぜなら入力可能な範囲を事前に知り、エラーを減らしたいからだ。	future

ステータスは3段階で定義。POが明示的にいると宣言されておらず、ざっくり判断するのに3つあればよいと判断

- Mvp:最初のリリースで含めるべき機能
- Nice to have:最初のリリースでは不要だが、あるとユーザー体験を向上させてくれるもの
- Future: 最初のリリースでは不要。またその後も優先しなくても良いが、あるとよりユーザビリティが増す。



- 看板ライクなツールでチームで議論しながら入力したものをcsv形式に変換
- 仕様書には記載されていないが、あるとよさそうな機能については、proposalの優先度を付与
- proposalの優先度のものについては仕様への提案としてまとめた(後述)

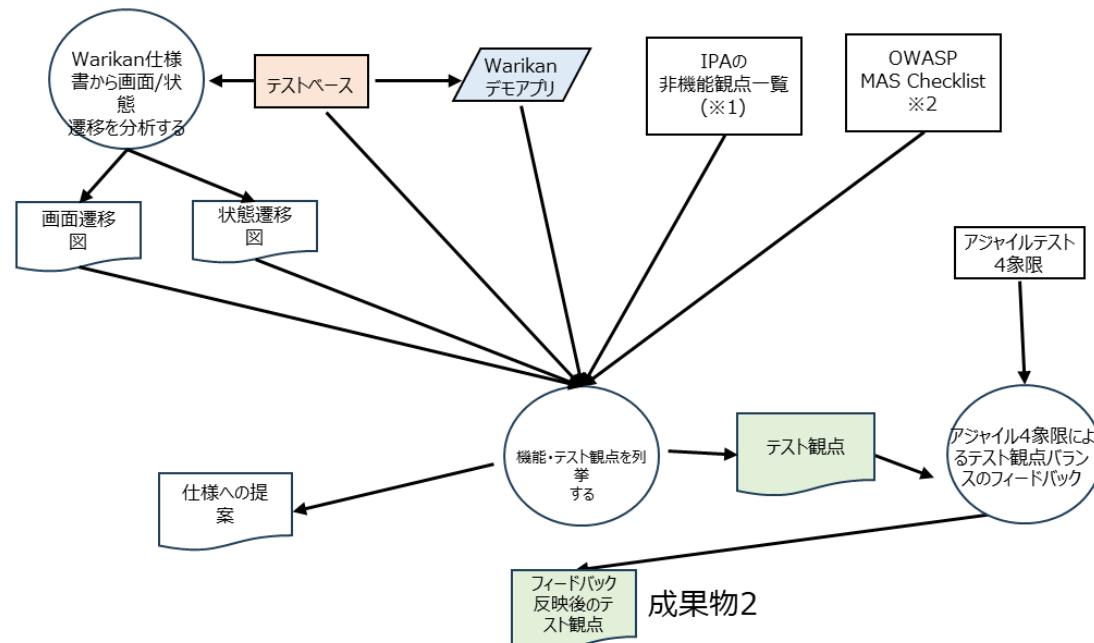
### テスト計画

テストスコープの定義や取るべきテストアプローチ、リスク、プロセスについて検討

# テストアーキテクチャ設計

テストアーキテクチャ設計に必要な情報を整理し、最終的に観点として集約

## 画面遷移図



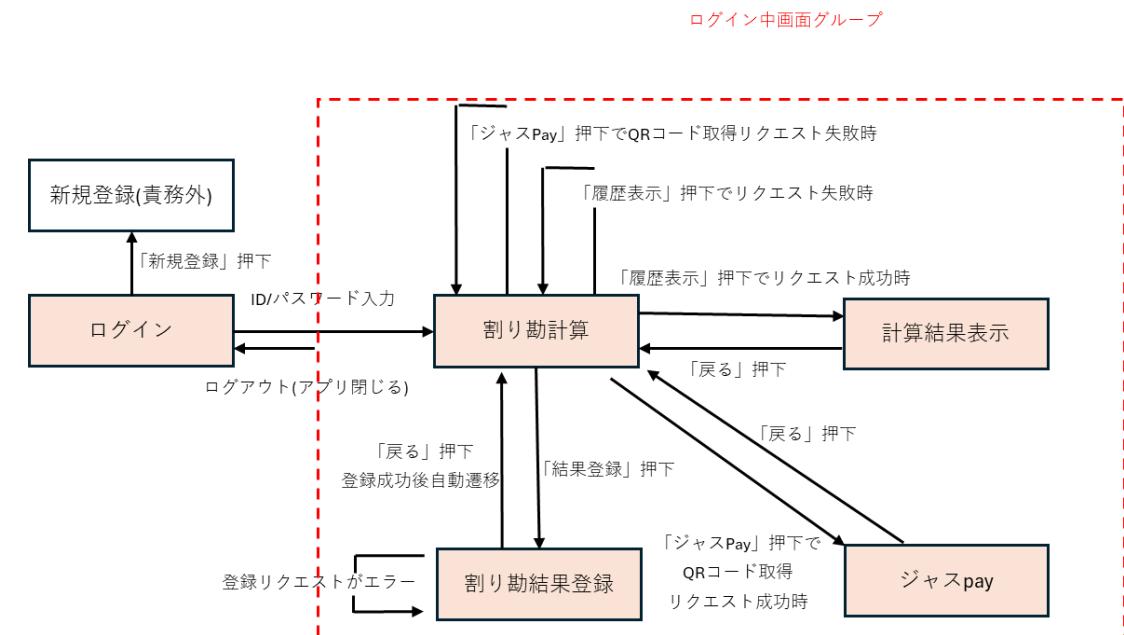
## Warikanアプリデモ版

- 実際に触って挙動を確認しながら、テストすべき項目を議論するのに役立てる。作成はAIで、工数はかけない

### 割り勘計算

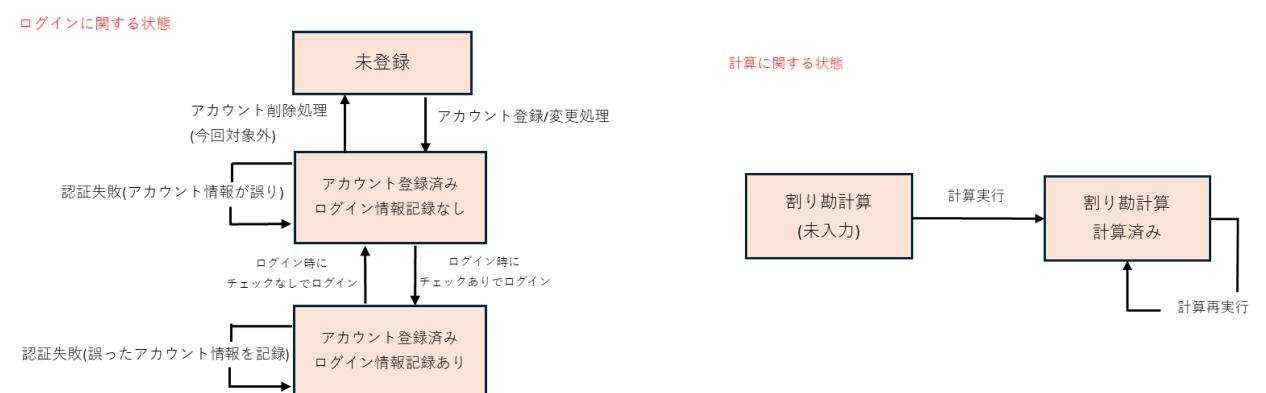
自分側の人数	<input type="text"/>
相手側の人数	<input type="text"/>
合計金額	<input type="text"/>
支払割合 (自分側)	<input type="range" value="50"/>
50% <input type="button" value="計算する"/>	
<input type="button" value="ジャスPay"/> <input type="button" value="結果を登録する"/> <input type="button" value="履歴を表示する"/>	

- 画面遷移を洗い出し、観点作成に役立てる

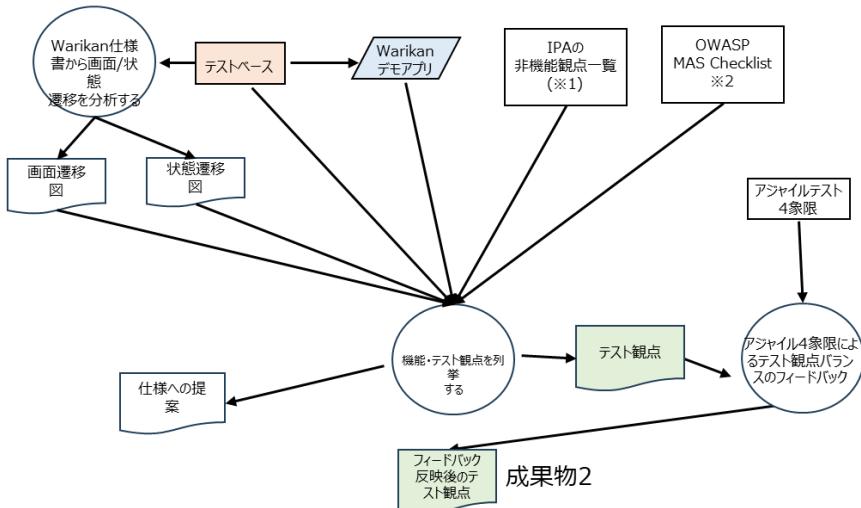


## 状態遷移図

- ログインに関する状態と、計算に関する状態を洗い出し、観点にフィードバック

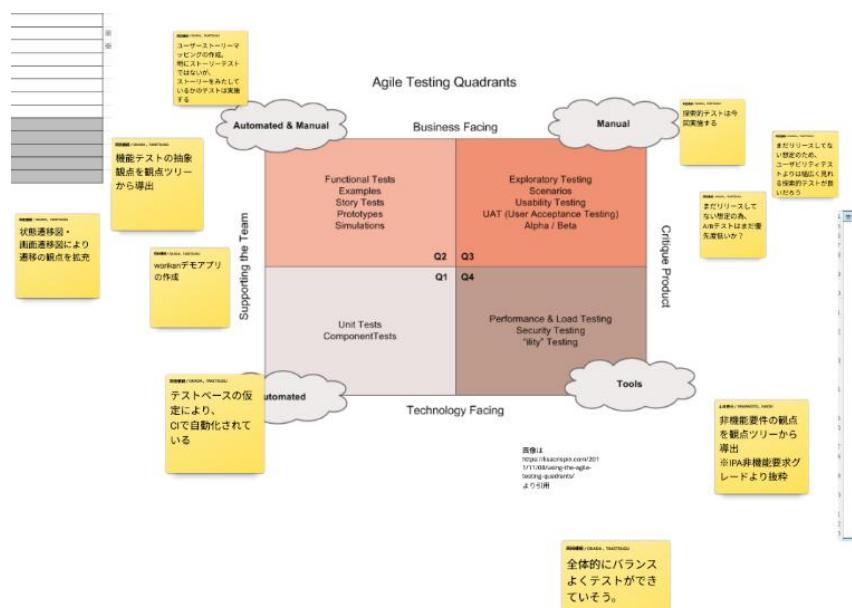


# テストアーキテクチャ設計



## アジャイルテスト4象限

- アジャイルで実施するテストを分類し  
バランスよくテストできることの確認を実施



## 仕様への提案(抜粋)

- ログインに関する状態と、計算に関する状態を洗い出し、観点にフィードバック

本シートでは、UXの観点で現状の仕様に追加した方が良いと考えた仕様の提案内容を記載します。				
項目番号	画面	内容	重要度	提案背景・理由
1	計算画面(計算実施前)	計算を実施する前に「結果を登録する」ボタンを押すと、エラーが表示される	nice to have	計算実施前にユーザが誤って「結果を登録する」ボタンを押した場合において、ユーザに不要な画面遷移をさせることがないので、混乱を招くことがない
2		自動で計算結果を登録するかどうかを設定できるチェックボックスを実装	nice to have	ユーザの計算結果登録が容易になる
3		備考欄で現在の入力文字数が(x/400)のように表示される	nice to have	ユーザが現在入力文字数を意識できる。
4		タイトルを入力できる	nice to have	ユーザは登録結果を見て、何の計算結果なのか一目で判別可能になる。
5		登録時に削除が発生する場合は確認を表示する	future	登録可能データは50件までのため、古いデータは現在自動で消えてしまう。 しかし、それがユーザにとって必要だった可能性もある。(50件までというのはユーザ走らないはず) 削除対象のタイトルを表示することで、ユーザは未認識のまま重要なデータを削除することを防げる。

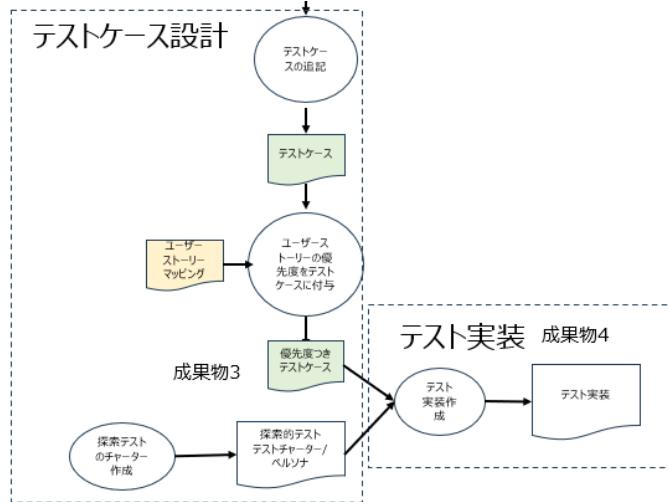
## テスト観点

- これまでの情報を集約し、観点を作成。必要なテストを明確化
  - 荒めの観点にとどめ、これ以上の詳細化はテストケース作成で行う

抽象テスト観点大項目	中項目(画面単位)	小項目	テスト観点
機能要件	計算結果表示	表示	パーツの色
			エラー通知
			パーツの初期表示
			計算結果表示
			表示状態保持
		入力	ボタン押下
			入力値チェック
			スライダーの入力
		処理	ログアウト処理
			サーバとの通信

# テストケース設計

HITACHI



テストケースとテスト実装をテストケースIDにて紐づける  
テストチャーターにて探索的テストを行う

## テストケース

抽象テスト観点大項	中項目(画面単位)	小項目	使用する共通機能	テスト観点	テストケース	テスト観点ID	関連するユーザーストーリーID	ユーザーストーリーの優先度	備考	テストケースID
				ページ表示	新規登録ボタン、ログインボタン、ユーザID入力フォーム、パスワード入力フォーム、チェックボックスが表示されていること	LGN-View-01		1 nice_to_have		LGN-View-01-TC

↓ テスト実装 ↓

ID	対応するテストケースのID	テストケースタイトル	前提条件	テストデータ	手順	期待結果	後処理	優先度
TD-NF-01		アプリケーションの運用時間・稼働率	・webクライアントを100台用意すること ・各クライアントで、ログイン・計算一登録・記録表示・ログアウトの流れを繰り返すスクリプトを作成しておくこと	以下が登録するデータ。 開催日: 8/18 開催時刻: 17:00 自分側の人数: 5人 相手側の人数: 5人 金額: 10000円 支払いの割合: (50:50) 自分側: 1000円/人 相手側: 1000円/人 おつり: 0円 備考: ロングラン用のデータ	1. それぞれのwebクライアントから24時間間隔でスクリプトを実行する	<ul style="list-style-type: none"> <li>定めた運用時間の間、サービス停止が発生することなく運用できる</li> <li>定めた稼働率(99.9%)の稼働時間以上正常に稼働する</li> <li>長時間の運用の中でレスポンスが長期化していないか</li> </ul>		高
NF-01-TC						<ul style="list-style-type: none"> <li>CPU利用率が80%以上になったり、60%以上の状態が1時間以上続くことがないか</li> <li>メモリ利用率が80%以上になったり、時間を経るごとに単調増加していく傾向がないか</li> <li>ディスク利用率が80%以上になったり、時間を経るごとに単調増加していく傾向がないか</li> </ul>		

↓ テストチャーター ↓

項目	名称	目的	ペルソナ	アプローチ	セッション時間	記録方法
1 割合スライダーの境界値操作と計算結果表示の検証	割合スライダーの最小・最大・中間値で正しい計算結果が表示されるか確認する	おごっていい所を見せたい利用者	エラーメッセージの内容とタイミング スライダー操作時のUI応答	ライダーを0%、50%、100%に設定し計算結果を確認 境界値附近(10%、90%)も試す 不正値(人数0、金額未入力)と組み合わせて挙動確認 観察ポイント: 計算結果が仕様通りか	45分 (実施30分 + 記録・振り返り15分)	テストメモ、スクリーンショット、必要に応じて画面録画
2 酔払い時のユーザビリティ検証	注意力・操作精度が低下した状態で誤操作や誤入力をしても誤計算や誤送信が防げるか。エラー内容を確認して正しい内容で操作ができるかを検証する	酔っぱらいの中年男性。老眼の為スマート画面が良く見えない。	誤タップからの復帰しやすさ	わざと行間違い・逆順入力など誤入力を スライダーやボタンを誤タップ 戻るボタン誤押下時のデータ保持状況を確認 観察ポイント: 誤入力防止機能や入力制限の有効性 エラー通知のわかりやすさ	45分 (実施30分 + 記録・振り返り15分)	テストメモ、スクリーンショット、必要に応じて画面録画

# テスト実装

HITACHI

- テスト観点とのトレーサビリティあり
- ユーザーストーリーマッピングに基づく優先度が分かる

テスト実装ID	対応するテストケースのID	テストケースタイトル	前提条件	テストデータ	手順	期待結果	後処理	優先
TD-LGN-View-01	LGN-View-01-TC	新規登録ボタン、ログインボタン、ユーザID入力フォーム、パスワード入力フォーム、チェックボックスが表示されていること	アプリが端末にインストールされており、まだ起動していないこと	なし	1. アプリを起動する。2. ログイン画面が表示されることを確認する。	ログイン画面に以下が全て表示されていること。・新規登録ボタン・ログインボタン・ユーザID入力フォーム・パスワード入力フォーム（マスク）	なし	nice_to_have
TD-LGN-Input-01	LGN-Input-01-TC	パスワード入力中に入力文字が●で表示されること	ログイン画面が表示されていること	パスワード=Abc123	1. ログイン画面のパスワード欄をタップする。2. パスワード欄に「Abc123」を入力する。	キーボードから入力した文字そのものは画面に表示されず、パスワード欄には「●」などのマスク文字のみが表示されること。	パスワード欄の入力をクリアする。	nice_to_have
TD-LGN-Error-01	LGN-Error-01-TC	エラーメッセージ内容にユーザが理解困難な用語（計算機の内部コード等）や短縮された言葉をメッセージ文に使用していないか	ログイン画面が表示されており、存在しないユーザIDまたは誤ったパスワードを入力できること	ユーザID=wrong@example.com、パスワード=Wrong1234	1. ログイン画面でユーザIDにwrong@example.com、パスワードにWrong1234を入力する。2. 「ログイン」ボタンを押下する。3. 表示されるエラーメッセージの文言を確認する。	エラーメッセージの本文に「ERR-001」などの内部コードや専門的すぎる略語がそのまま表示されないこと。一般ユーザが意味を理解できる自然な日文で原因と状況が説明されていること。	エラー・ポップアップを閉じ、入力値をクリアする。	nice_to_have
TD-LGN-Error-02	LGN-Error-02-TC	エラーメッセージにはユーザの取るべき対処が書かれているか。またユーザに対処可能な内容か	ログイン画面または割り勘計算画面が表示されており、入力エラーと通信エラーをユーザ操作で再現できること。端末のネットワーク接続をON/OFF切り替え可能であること。	入力エラー用：ユーザID空欄または人數=0、金額=0通信エラー用：ネットワーク=OFF、有効なユーザID/パスワード	1. ネットワークONの状態で、ログイン画面または割り勘計算画面でユーザID未入力や人數0など不正値を設定する。2. 「ログイン」または「計算する」ボタンを押下し、入力エラーを発生させる。3. 表示されたエラーメッセージを確認する。4. 次に端末のネットワークをOFFにする。5. 正しい値を入力した状態で「ログイン」「結果登録」「履歴表示」など通信を伴うボタンを押下し、通信エラーを発生させる。6. 表示されたエラーメッセージを確認する。	入力エラー時：・「1~99の数値を入力してください」など、ユーザが次に行うべき具体的な操作が記載されていること。・記載された操作がユーザ権限内で実行可能な内容であること。通信エラー時：・「通信環境を確認して再度お試しください」「電波状況を確認してください」などの対処内容が記載されていること。・ユーザがその指示どおりに現実的に操作できる内容であること。	ネットワークをONに戻し、画面を通常状態に戻す。	nice_to_have
TD-LGN-Error-03	LGN-Error-03-TC	エラーメッセージは重要度がわかるようなデザインになっているか	ログイン画面が表示されており、正しくないケレデンシャルを入力した状態を再現できること	ユーザID=wrong@example.com、パスワード=Wrong1234	1. ログイン画面でユーザIDにwrong@example.com、パスワードにWrong1234を入力する。2. 「ログイン」ボタンを押下する。3. 表示されるエラー通知の背景色、アイコン、文字の強調などのデザインを確認する。	ログインエラーのエラーメッセージが、通常の情報よりも目立つ色やアイコン・フォント強調で表示され、ユーザが「重要なエラー」であると視覚的に認識できること。	エラー・ポップアップを閉じる。	nice_to_have
TD-LGN-Error-04	LGN-Error-04-TC	ポップアップ外側をタップして消去できること	いずれかの操作でエラー通知ポップアップが表示されている状態であること	なし	1. ログイン失敗などの操作でエラー通知ポップアップを表示する。2. ポップアップ内のボタンには触れず、ポップアップ外側（背景部分）をタップする。	ポップアップ外側をタップした操作でエラー通知ポップアップが閉じられ、直前の画面（ログイン画面など）に戻ること。	画面が正常に操作可能な状態であることを軽く確認する。	nice_to_have