Картина, която съдържа текст, графична колекция

Описанието е генерирано автоматичноТЕХНИЧЕСКИ УНИВЕРСИТЕТ – ВАРНА

Факултет по изчислителна техника и автоматизация

Катедра „Софтуерни и интернет технологии“

**ПРОЕКТ**

По дисциплина: Управление на софтуерни проекти

Тема: Система за управление на недвижими имоти

|  |
| --- |
| **Изготвили:** |
| Антон Антонов-  Мартин Атанасов-  Сонер Ибрям- |
| **Проверил:**  / / |

# Въведение

Тази документация служи както за описание на техническия аспект на проекта, така и за отчитане на управленските подходи, използвани по време на неговото изпълнение. Основна цел е не само реализацията на функционален продукт, но и практическото приложение на хибридни модели за разработка.

**Цел на документа:**

Да се опишат процесите по управление на софтуерния проект "ИмотБГ" с акцент върху използваните инструменти за управление (MS Project, Trello, GitHub) и методологиите, приложени в съответствие с изискванията на дисциплината "Управление на софтуерни проекти".

**Обхват на проекта:**

Разработка на уеб приложение за търсене, публикуване и управление на обяви за недвижими имоти с фронтенд на HTML/CSS/JavaScript и локално съхранение на данни (localStorage).

# Инструменти за управление на проекта

Подборът на инструменти беше направен с цел постигане на баланс между стратегическо планиране и ежедневна оперативна ефективност. Беше важно всеки инструмент да допринася за повишаване на прозрачността, отчетността и възможността за бърза реакция при възникване на проблеми.

2.1. **MS Project**

**Роля в проекта:**

* Използван за **глобално планиране** и проследяване на критичния път.
* Помогна за визуализация на зависимостите между задачите и изчисляване на сроковете.

**Конфигурация:**

* **График на Гант:** Показва фазите на проекта (инициация, планиране, изпълнение, тестване).
* **Ресурси:** Назначаване на екипа към конкретни задачи.

**Предимства:**

* Точно предвиждане на сроковете.
* Възможност за симулации на забавяния.

**Недостатъци:**

* Ограничена колаборация в реално време.
* Сложен за бързи промени по време на спринтове.

2.2. **Trello**

**Роля в проекта:**

* Използван за **управление на дневните задачи** и Agile процеси.
* Интегриран с GitHub за автоматично обновяване на статусите.

**Настройки:**

* **Kanban дъска:**
  + *To Do*: Задачи за текущия спринт.
  + *In Progress*: Активни задачи.
  + *Code Review*: Очакващи ревю.
  + *Done*: Завършени.
* **Power-Ups:**
  + GitHub Integration – автоматично свързване на PR с картички в Trello.

**Предимства:**

* Интуитивен и лесен за използване.
* Подходящ за бързи промени по време на спринтове.

**Недостатъци:**

* Ограничени възможности за отчети.

2.3. **GitHub**

**Роля в проекта:**

* **Хостинг на кода** и управление на версиите.
* **GitHub Projects** за проследяване на задачи (като допълнение към Trello).

**Конфигурация:**

* **Automated Workflows:**
  + При създаване на PR се премества в колона "Review".
  + При затваряне на PR се маркира като "Done".
* **Метрики:**
  + Покритие на код (чрез GitHub Actions).
  + Брой решени issues/седмица.

**Предимства:**

* Пълна интеграция между кода и управлението на задачите.
* Вградени инструменти за код ревю.

**Недостатъци:**

* Ограничена гъвкавост в custom workflows спрямо Trello.

# 3. Сравнение на инструментите

| **Инструмент** | **Предимства** | **Недостатъци** | **Най-добра употреба в проекта** |
| --- | --- | --- | --- |
| **MS Project** | Точно планиране, проследяване на ресурси | Сложен за бързи промени | Глобално планиране и фазови анализи |
| **Trello** | Лесна колаборация, гъвкавост | Липса на разширени отчети | Дневни задачи и Agile управление |
| **GitHub** | Интеграция с кода, автоматизации | Ограничени опции за управление на задачи | Проследяване на технически задачи |

# 4. Методология за разработка (Hybrid Agile)

**Комбиниран подход:**

* **Scrum** за ежедневни задачи (чрез Trello) - Scrum спринтовете от по една седмица бяха достатъчно кратки за поддържане на интензивна итерация, а ежедневните срещи способстваха за бързо идентифициране на блокиращи проблеми.
* **Waterfall елементи** за високо ниво на планиране (чрез MS Project).

**Процеси:**

1. **Инициация (Waterfall):**
   * Определяне на изискванията и срокове в MS Project.
2. **Изпълнение (Scrum):**
   * 1-седмични спринтове с ежедневни standup-и.
   * Управление на задачите в Trello.
3. **Мониторинг (GitHub):**
   * Проследяване на напредъка чрез GitHub Insights.

**Управление на промените:**

* Промени в изискванията се документират като нови картички в Trello с етикет *Change Request*.

# 5. Управление на рискове

Идентифицирането на рискове беше извършено в началото на проекта, като част от фазата "Инициация". Всеки риск бе оценен по вероятност и въздействие. Използването на GitHub като централен хранилищен механизъм елиминира риска от загуба на код и данни.Коментарите и етикетите в Trello позволиха документиране на промените в реално време, което намали недоразуменията при комуникацията.

**Идентифицирани рискове:**

| **Риск** | **Инструмент за митигация** | **Действие** |
| --- | --- | --- |
| Забавяне на критичния път | MS Project (симулации) | Преразпределяне на ресурси |
| Губи на данни (localStorage) | GitHub (backup на код) | Редни експорти на данни |
| Комуникационни бариери | Trello (коментари в задачи) | Редни sync срещи |

# 6. Изводи и препоръки

**Резултати:**

* **MS Project** беше незаменим за стратегическо планиране.
* **Trello** улесни Agile процесите, но отчетите се генерираха ръчно.
* **GitHub** спести време в интеграциите, но беше недостатъчен за управление на целите.

**Препоръки:**

* Да се използва **Trello + GitHub** за малки екипи.
* Да се добави **Jira** за по-сложни отчети в бъдещи проекти.

**Оценка на екипа:**

* **Ефективност:** 8/10 – комбинацията от инструменти покри всички нужди.
* **Урок:** Важно е ясно да се дефинират ролите на всеки инструмент още в началото.

# 7. Архитектура на приложението

Всяка HTML страница в проекта е отделена по функционалност, което улеснява поддръжката и разширението на кода. CSS файловете са групирани в папка, а скриптовете са модулни, което позволява лесно рефакториране.

Изборът на localStorage като основен механизъм за съхранение на данни е обусловен от учебния характер на проекта – така се избягва нуждата от конфигуриране на сървърна инфраструктура.

* 7.1 Структура:  
  - **proekt.html** – начална страница със списък от обяви  
  - **add-property.html** – форма за добавяне на нов имот  
  - **property-details.html** – изглед на детайлите на обява
* 7.2 Технологии:  
  - **HTML5, CSS3, JavaScript (Vanilla)**- Без backend – използва се **localStorage** за временно съхранение  
  - Всички данни се съхраняват и зареждат в браузъра на потребителя

# 8. Основни функционалности

|  |  |  |
| --- | --- | --- |
| Функция | Файл/Скрипт | Описание |
| Добавяне на обява | add-property.html + add.js | Потребителят въвежда информация, която се записва в localStorage. |
| Преглед на обяви | proekt.html + main.js | Извличане на обявите от localStorage и рендериране в списък. |
| Детайлен изглед | property-details.html | Показва допълнителна информация за конкретен имот. |