

ניהול פרויקטי תוכנה - פרויקט גמר

כל קבוצה תגיש תיק פרויקט שיכיל את הנושאים הבאים המבוססים על התרגילים הקודמים ושימוש בביבליוגרפיה.

יש לבחור באותה אפשרות, A או B, כמו בתרגילים.

A. נושא פרויקט הגמר:

1. תיאור הפרויקט (תרגיל 1).

2. Market Requirements Summary

- א. הגדרת קהל היעד (תרגיל 1)
- ב. פרסונות, ניתן להשתמש בפרופילי facebook, LinkedIn כמייצגים (תרגיל 1).
- ג. ניתוח תחרות (תרגיל 1).
- ד. הסבירו מדוע קהל היעד ירצה להשתמש בפרויקט שלכם.

3. User Requirements Summary

- א. סיפורי משתמש, user stories (תרגיל 1).
- ב. מאפיינים, features וסדרי עדיפויות: חובה, רצוי, עתידי (תרגיל 1).
- ג. תכנון ממשק, storyboard (תרגיל 1).
- ד. מסכי משתמש, UI (תרגיל 1).
- ה. חווית משתמש, UX ולינק לאב-טיפוס ב-inVision (תרגיל 1).
- ו. סקר משתמשים (תרגיל 3).

4. System Requirements Summary

- א. תיאור הארכיטקטורה הכללית (תרגיל 2).
- ב. תרשים של הארכיטקטורה ותיאור של הרכיבים העיקריים (תרגיל 2).
- ג. תרשים עבור ה-flows העיקריים במערכת.
- ד. טבלאות עיקריות ושדות עיקריים שישמרו בבסיס הנתונים (תרגיל 2).
- ה. תיאור הטכנולוגיות שבהן הפרויקט יפותח: שפות תכנות, ספריות, רכיבים מוכנים, API's, סביבות פיתוח, כלי הפיתוח (תרגיל 2).
- ו. תיאור המחלקות העיקריות, הפונקציות והשדות העיקריים (תרגיל 2).
- ז. תרשים מחלקות עיקריות, class diagram (תרגיל 2).

5. Testing and QA Summary

- א. תיאור של תהליך הבדיקות.
- ב. תיאור של סביבת הבדיקות, חומרה, תוכנה, דרישות נוספות.

- ג. תסריטי בדיקות עבור סיפור המשתמש וה-flows העיקריים.
- ד. תיאור שלבי הבדיקה והתוצאות הצפויות.

6. Project Management Methodology and Tools

- א. תיאור מתודולוגיה לניהול הפרויקט וכלי ניהול. בחרו במתודולוגיה אחת למשל מבין Agile, Rapid Prototyping, XP, Scrum והסבירו את בחירתכם ויתרונות המתודולוגיה עבור הפרויקט שלכם.
- ב. ניתוח משאבים, זמן, היקף (תרגיל 3).
- ג. פתחו פרויקט חינמי ב-GitHub עבור הפרויקט שלכם.
- ד. הסבירו מהם יכולות ניהול הפרויקט הקיימות ב-GitHub וכיצד ניתן להשתמש בהם לשיתוף קוד, code reviews וניהול הפרויקט שלכם.
- ה. בחרו בכלי נוסף מבין הכלים המתאימים לניהול ופיתוח הפרויקט בקבוצה שנלמדו בשיעור למשל Jira, Slack והירשמו לניסיון חינמי בכלי ופתחו פרויקט חדש עם משתמשים. הסבירו את היכולות, יתרונות וחסרונות של הכלי, וצרפו צילומי מסך.

7. ביבליוגרפיה.

B. נושא רכב אוטונומי

1. תיאור הפרויקט

א. הסבירו את היתרונות השונים של רכבים אוטונומיים (תרגיל 1).

2. Market Requirements Summary

א. הגדרת קהל היעד (תרגיל 1).

ב. ניתוח תחרות, כולל דרגת אוטונומיות מ-0 עד-5 של המתחרים (תרגיל 1).

היעזרו בלינק למצגת המכילה היקף קהל יעד, פירוט דרגות אוטונומיה שונות, אפיון מתחרים, ושאלות ותשובות, הנמצא במודל.

3. User Requirements Summary

א. השוו בין מסכי המשתמש והממשק לנוסע ברכב האוטונומי של Tesla ו-Uber. צרפו צילומי מסך. מהם היתרונות והחסרונות? מה הייתם משנים?

ב. כיצד באה לידי ביטוי דרגת האוטונומיות של הרכבים של Tesla ו-Uber בממשק לנוסע וחווית המשתמש.

ג. סקר משתמשים (תרגיל 3).

4. System Requirements Summary

יש לבחור בארכיטקטורה אחת מבין הארכיטקטורות של החברות הבאות:

Comma.ai, PolySync (OSSC), Autoware, nVidia

א. תיאור הארכיטקטורה הכללי.

ב. תרשים של הארכיטקטורה ותיאור מפורט של הרכיבים העיקריים.

ג. תרשים עבור כל רכיב תוכנה במערכת.

ד. תיאור הטכנולוגיות שבהן הפרויקט יפותח: שפות תכנות, ספריות, רכיבים

מוכנים, API's, סביבות פיתוח, כלי הפיתוח.

ה. מודולים: תיאור כיצד רכיבי התוכנה פועלים והקשרים ביניהם. פונקציונאליות

עיקרית של רכיבי התוכנה. יש לכלול את כל ה-SDK's.

היעזרו בתרגיל 2 ובמסמך המתאר את הארכיטקטורה של nVidia הנמצא במודל.

5. Testing and QA Summary

- א. תיאור של תהליך הבדיקות.
- ב. תיאור של סביבת הבדיקות, חומרה, תוכנה, דרישות נוספות.
- ג. תסריטי בדיקות.
- ד. תיאור שלבי הבדיקה והתוצאות הצפויות.

6. Project Management Methodology and Tools

- א. תיאור מתודולוגיה לניהול הפרויקט וכלי ניהול. בחרו במתודולוגיה אחת למשל מבין Agile, Rapid Prototyping, XP, Scrum והסבירו את בחירתכם ויתרונות המתודולוגיה עבור הפרויקט שלכם.
- ב. ניתוח משאבים, זמן, היקף.
- ג. פתחו פרויקט חינוכי ב-GitHub עבור הפרויקט שלכם.
- ד. הסבירו מהם יכולות ניהול הפרויקט הקיימות ב-GitHub וכיצד ניתן להשתמש בהם לשיתוף קוד, code reviews וניהול הפרויקט שלכם.
- ה. בחרו בכלי נוסף מבין הכלים המתאימים לניהול ופיתוח הפרויקט בקבוצה שנלמדו בשיעור למשל Jira, Slack והירשמו לניסיון חינוכי בכלי ופתחו פרויקט חדש עם משתמשים. הסבירו את היכולות, יתרונות וחסרונות של הכלי, וצרפו צילומי מסך.

7. ביבליוגרפיה.

References

- [AWS, 2016] AWS (2016). AWS architecture center.
<https://aws.amazon.com/architecture>
- [Bell, 2013] Bell, P. (2013). Managing software development. Google for Entrepreneurs and General Assembly.
- [Beyer et al., 2016] Beyer, B., Jones, C., Petoff, J., and Richard, N. M. (2016). Site Reliability Engineering: How Google Runs Production Systems. O'Reilly Media.
- [Direkova, 2015] Direkova, N. (2015). Google design sprint.
<https://developers.google.com/design-sprint/downloads/DesignSprintMethods.pdf>
- [Docker, 2016] Docker (2016). Docker containers. <https://www.docker.com>.
- [Drori, 2016] Drori, I. (2016). Software project management landscape.
www.idrori.com/courses/spm-landscape.pdf.
- [Evans and Basole, 2016] Evans, P. R. and Basole, R. C. (2016). Revealing the API ecosystem and enterprise strategy via visual analytics. *Communications of the ACM*, 59(2).
- [Eyal, 2014] Eyal, N. (2014). *Hooked: How to build habit-forming products*. Penguin Canada.
- [Feitelson et al., 2013] Feitelson, D., Frachtenberg, E., and Beck, K. (2013). Development and deployment at Facebook. *IEEE Journal Internet Computing*, 17(4):8–17.
- [GitHub, 2016] GitHub (2016). GitHub new features.
<https://github.com/blog/category/ship>.
- [Granka, 2013] Granka, L. (2013). 5 product research tools. Google for Entrepreneurs and General Assembly.
- [inVision, 2016] inVision (2016). inVision application.
<https://www.invisionapp.com>.
- [Larman and Basili, 2003] Larman, C. and Basili, V. R. (2003). Iterative and incremental developments: A brief history. IEEE Computer Society.
- [McCandless, 2016] McCandless, D. (2016). Codebase lines of code visualization. <http://www.informationisbeautiful.net/million-lines-of-code>
- [Potvin and Levenberg, 2016] Potvin, R. and Levenberg, J. (2016). Why Google stores billions of lines of code in a single repository. *Communications of the ACM*, 59(7).
- [Savor et al., 2016] Savor, T., Douglas, M., Gentili, M., Williams, L., Beck, K., and Stumm, M. (2016). Continuous deployment at Facebook and OANDA. In *International Conference on Software Engineering*, pages 21–30. ACM.
- [@Scale, 2016] @Scale (2016). @scale conference.
<http://www.atscaleconference.com>.
- [Schwarz, 2015] Schwarz, Roger (2015). How to Design an Agenda for an Effective Meeting, *Harvard Business Review*.
- [UserOnBoard, 2016] UserOnBoard (2016). User on-boarding.
<https://www.useronboard.com>