SmartNotes

Smart dokumentredigerare med stöd för ljudinmatning, matematiska formler samt lagringsmöjligheter på Google Drive.

Smart document editor with support for speech-to-text, mathematical formulas and ability to save to Google Drive.

David Guirguis, Joel Vik, Erik Hellenberg, Anton Espling, Samsom Dewitsegid, Henric Andersson

https://www.github.com/SmartNotes/

Mjukvarukonstruktion, projektkurs inom Datateknik
Grundnivå, 7.5 hp
Handledare: Jonas Willén
Examinator: Reine Bergström

KTH
Skolan för kemi, bioteknologi och hälsa
141 52 Huddinge, Sverige

Sammanfattning

SmartNotes är en textredigerare som genererar förslag och därefter funktioner eller symboler baserat på valt förslag. Tal-till-text implementerades för att förbättra användarvänligheten och generera text, symboler och funktioner med endast rösten som verktyg. Applikationen skapades med syftet att förenkla generering av tecken och funktioner som inte finns på tangentbordet eller kräver flera tidskrävande steg i de idag populära dokumentredigerarna som Google Documents och Microsoft Word. Utvecklandet av SmartNotes gjordes i JavaScript, huvudsakligen med ramverket React. CircleCI användes för kontinuerlig integration och applikationen driftsattes med AWS.

Nyckelord

Textredigerare, automatiska inmatningsförslag, teckengenerering, matematik, tal-till-text, Google Drive-integration, JavaScript, TypeScript, React, Node, kontinuerlig integration, kontinuerlig driftsättning.

Abstract

SmartNotes is a text editor that generates suggestions and features or symbols based on the selected suggestion. Speech to text was implemented to improve user-friendliness and generate text, symbols and features using only voice as a tool. The application was created with the aim of simplifying the generation of characters and features that are not on the keyboard or requires several time-consuming steps in popular document editors like Google Documents and Microsoft Word. The development of SmartNotes was done in JavaScript, mainly with the React framework. CircleCI was used for continuous integration and the application was deployed with AWS.

Keywords

Text editor, autocomplete, character generation, mathematics, speech-to-text, Google Drive integration, JavaScript, TypeScript, React, Node, continuous integration, continuous deployment.

Förord

Den applikation gjordes möjlig av sex hårt arbetande studenter vid KTH tillsammans med handledning av Jonas Willén. Tack till de som testat, varit med i undersökning och i övrig bidragit till denna produkt.

Ordförklaring

API - Application Programming Interface

Användarvänlig - Definierat enligt “The Research-Based Web Design & Usability Guidelines, Enlarged/Expanded edition” [1]

GIT - Versionshanteringsprogram som bland annat används för att dela kod

Innehållsförteckning

1 Inledning

1.1 Problemformulering

I dag finns det ingen dokumentredigerare som inkluderar de komplexa verktygen som täcker alla behov för matematiskt skrivande, vilket kan handla om allt från matematiska avhandlingar till enkla anteckningar. Avsikten med SmartNotes är att skapa en användarvänlig dokumentredigerare som underlättar skrivandet av just dessa dokument.

1.2 Målsättning

Målet är att använda minst två nya tekniker som projektmedlemmarna inte har arbetat med tidigare. De valda teknikerna är tal-till-text samt kontinuerlig integration. Dessutom ska produkten ha en arkitektur av hög kvalitet. Målet bortom det att använda två nya tekniker är att utveckla en produkt som fyller det tidigare nämnda behovet och därmed förenklar daglig dokumentredigering.

SmartNotes syftar till att erbjuda en lättillgänglig lösning som både är avgiftsfri och kräver färre steg för infogning av specialtecken och ekvationer jämfört med Microsoft Word, *se kap. 2.3.2*.

1.3 Avgränsningar

1.3.1 Teamstorlek

Programmet SmartNotes utvecklas i ett team på sex personer. Funktionerna begränsas till hur mycket denna grupp hinner göra inom den tidsram som finns för kursen Mjukvarukonstruktion HI1036 vid KTH [2].

1.3.2 Tid

Gruppen har för hela arbetet ca 170 timmar per person, d.v.s. 1020 arbetstimmar fördelade på sex gruppmedlemmar, att utveckla mjukvaran samt skriva en vetenskaplig rapport som beskriver denna.

1.3.3 Implementation

En möjlig funktion och implementation skulle kunna vara att direkt i SmartNotes kunna beräkna de matematiska komplexa uttrycken och funktionerna. Detta är något som har bortprioriteras för att bli klara inom tidsramarna med projektet. Att flertalet användare samtidigt kan skriva och integrera med SmartNotes är för vidare en utvecklingsmöjlighet i framtiden. Dessutom kommer det inte att vara möjligt att infoga bilder eller annan grafik i SmartNotes.

2 Bakgrund och Teori

I det här kapitlet diskuteras bakgrund till problemet som är nödvändigt för att förstå arbetets syfte.

2.1 Bakgrund

Google Documents, som är en kostnadsfri dokumentredigerare, har många olika funktioner för att skapa och redigera dokument. En av de funktionerna är att infoga vissa specialtecken, men det kan upplevas som besvärligt då det kräver många steg, *se kap. 3.2.1*. Autocomplete på specialtecken finns inte och det är en komplicerad väg att välja bland menyer för att hitta rätt tecken. Det finns externt skapade tillägg som utvecklats av tredje part vars mål är att lösa detta problem [3], men de är svåra att hitta och inte alltid enkla att använda [4].

Microsoft Word är en annan populär ordbehandlare. Microsoft tar dock ut en avgift i form av en prenumeration för användandet av produkten.

2.2 Fallstudie

En fallstudie utfördes under projektets gång för att jämföra effektiviteten av att använda kända dokumentredigerare för infogning av specialtecken, för att sedan jämföra resultatet med SmartNotes.

2.3.1 Google Documents

För att införa ett specialtecken i Google Documents behöver en användare utföra fem steg. Stegen som följer är:

1. Välj “Infoga” från menyraden.
2. Välj “Specialtecken” från undermenyn.
3. Välj kategori för önskat tecken.
4. Leta upp önskat tecken i listan över specialtecken.
5. Välj önskat tecken.

[5]

2.3.2 Microsoft Word

För att införa ett specialtecken i Microsoft Word behöver en användare utföra fyra till sex steg. Stegen som följer är:

1. Välj “Infoga” från menyraden.
2. Välj “Symbol” från undermenyn.
3. Leta upp önskat tecken i listan över specialtecken.
4. Om önskat tecken hittas, välj tecknet. Annars, välj “Fler symboler...”.
5. Leta upp önskat tecken i den utökade listan över specialtecken.
6. Välj önskat tecken.

[6]

2.3 Teori

Genom projektet användes olika utvecklingsverktyg för att på ett effektivt sätt skapa de delar av mjukvaran som var tänkt.

2.3.1 Kontinuerlig integration

Kontinuerlig integration är en rad olika principer som bygger på att kod laddas upp till ett sk. *repository*. Ofta testas denna kod direkt på en server som bygger och kompilerar kodbasen. Testerna är menade att utföras automatiskt och ta reda på om den senaste ändringen i koden orsakar problem i applikationen [7].

Med kontinuerlig integration är det en fördel och faktiskt rekommenderat att inte skriva kod i en lokal miljö och när den är färdig, laddas upp till *master-branchen*, utan i stället laddar upp all kod som skrivs, direkt till *master-branchen*. Detta eftersom verktygen för kontinuerlig integration gör det lämpligt att skriva och få små ändringar testade direkt [8].

2.3.2 Speech-to-text

Speech-to-text-integration innebär att användare kan kontrollera datorfunktioner och kan diktera text med rösten. Kort och simpelt förklarat består mjukvaran av två komponenter, en komponent som bearbetar de akustiska signalerna som fångas upp av mikrofonen och den andra komponenten tolkar dessa signaler och transformerar de till ord [9].

2.3.3 Utvecklingsverktyg

2.3.2.1 **CircleCI**

CircleCI är det kontinuerlig integrationsverktyg som används i detta projekt. CircleCI integreras genom GitHub och varje gång kod uppdateras till SmartNotes repository kör CircleCI alla de fördefinierade testerna, som skrivits för denna mjukvara, på en ny container eller virtuell maskin, för att därefter meddela om testerna lyckats eller ej [10].

2.3.4 LaTeX

Ett av de mest populära typsättningssystemen för matematiska dokument är LaTeX. LaTeX är ett märkspråk som genererar dokument baserat på beskrivningskod, vilket innebär att resultatversionen inte är direkt modifierbar. [11]

LaTeX utgår från principen att olika författare av dokument ska kunna fokusera på innehållet och överlåta designen av dokumentet till någon annan [12].

2.3.5 React

React är ett deklarativt komponentbaserat JavaScript-bibliotek utvecklat för att skapa avancerade, interaktiva användargränssnitt [13]. För att få en bättre strukturerad kodbas används React istället för ren JavaScript eftersom det erbjuder större struktureringsmöjligheter i form av komponenter.

2.3.6 TypeScript

TypeScript är skapat i open-source och underhålls av Microsoft. TypeScript omvandlas till ren JavaScript-kod vid kompilering. Utan denna omvandling kan inte webbrowsern läsa koden och därmed inte heller visa upp innehållet. TypeScript räknas inte som ett objektorienterat språk vilket java gör, utan räknas som ett scripting language. Typescript har “static typing”, ger support till moduler, stödjer användandet av interface-implementation med mera. TypeScript är kort beskrivet JavaScript men med extra funktioner [14].

2.3.7 Pusher

Pusher Channels är ett eventbaserat API som använder websockets och HTTP för att kommunicera över kanaler i realtid mellan applikationer, servrar och enheter [15]. Publika-, privata- eller närvarokanaler kan användas beroende på vad för data som ska skickas [16][17][18].

2.3.8 AWS Amplify

AWS Amplify är en utvecklingsplattform för att bygga säkra och skalbara mobil- och webbapplikationer. Det gör det enkelt att autentisera användare och lagra data på ett säkert sätt. Enkelt att integrera maskininlärning, analysera applikations mätningar och exekvera *server-side* kod. AWS Amplify täcker det kompletta arbetsflödet för mobilapplikationer från versionskontroll till kod testning till produktionsdistribution och det skalar enkelt med ens företag med tusentals användare ända upp till miljontals. Amplify-biblioteken och CLI, som ingår i amplify framework, är open source och erbjuder ett pluggbart gränssnitt som gör att du kan anpassa och skapa dina egna plugins. [19]

2.3.9 AWS EC2

Amazon Elastic Compute Cloud (Amazon EC2) är en webbservice som förser säkra och i storlek modifierbara molnlösningar. EC2 är designat för att göra webbaserade projekt enklare att skapa för utvecklare. Dess simpla webbtjänst-interface tillåter utvecklare att erhålla och konfigurera kapacitet med minimala problem. Dessutom förser tjänsten fullständig kontroll över dataresurser samt ett sätt att använda Amazons beprövade tekniska miljö. [20]

3 Metoder och resultat

Genom det här kapitlet beskrivs hur litteraturstudien utfördes och hur den bidrog till att göra kvalitativa val. Dessutom framställs vilka metoder som använts genom projektet.

3.1 Metod

3.1.1 Förstudie

För att kunna göra kvalitativa val genomfördes en litteraturstudie.

Även en marknadsundersökning utfördes där de vanligaste textredigerarna testades grundligt utifrån hur användarvänlig funktionaliteten gällande generering av specialtecken var. Fallstudier gjordes för Google Documents och Microsoft Word 2019 *se kap. 2.2*.

Fallstudien visade att antalet steg det tog att generera specialtecken eller matematiska funktioner inte var få nog för att processen ska klassas som användarvänlig i de undersökta applikationerna [1].

3.1.2 Utveckling av webbapplikation

Genom projektet användes olika språk och ramverk för de olika delarna. De olika delarna i projektet går att dela upp på följande vis:

- Frontend
- Backend
- Kommunikation mellan Frontend och Backend
- Kontinuerlig Integration
- Produktion

3.1.2.1 Frontend

Utvecklingen av Frontend valdes att skrivas i JS baserat på gruppens tidigare erfarenheter. React valdes istället för ren JavaScript då det med hjälp av de funktionerna som medförs gjorde utvecklingen av produkten snabbare och mer effektiv. Dessutom är React väldigt användbart för att få bra struktur på programmet.

Alternativt behöver man inte använda react, men det som blir lidande är strukturen av programmet. Ett snarlikt alternativ till React är Angular i hänsyn till användningsområdet. React valdes av den anledningen att projektgruppen hade tidigare kunskaper om React.

3.1.2.2 Backend

Med fokus på JavaScript gjordes backend med NodeJS. För att kunna strukturera upp koden och möjliggöra en mer objektorienterad struktur skrevs den i Typescript.

3.1.2.3 Kommunikation mellan backend och frontend

React-frontenden kommunicerar med NodeJS-backenden genom Pusher Channels API. Pusher för att tidigare erfarenheter med det fanns inom gruppen. Kanaler för kommunikation mellan klienterna och servern används för att skicka gjorda anteckningar och begära API anrop av backenden för autentisering och uppladdning av anteckningarna till Google Drive.

3.1.2.4 Kontinuerlig Integration

CircleCi valdes för att sätta upp kontinuerlig integration eftersom det underlättar integrationen med GitHub, jämfört med andra plattformar såsom Jenkins. Testerna körs i en Linuxbaserad Dockerkontainer med Node 8. (Bilaga 1)

3.1.2.5 Produktion

Produktionssättningen gjordes i Amazon AWS, då de erbjuder gratis hostning och kontinuerlig driftsättning.

3.1.2.5.1 Frontend

Frontendproduktion körs i AWS Amplify. För att ändringar eller uppdateringar ska lanseras utan komplikationer sattes det upp med kontinuerlig driftsättning.

3.1.2.5.2 Backend

Backendproduktion körs i AWS EC2, en virtuell maskin med Linux. För motivering av valet kring EC2 se kap. 4.2.

3.2 Resultat

SmartNotes är en dokumentredigerare som bland annat klarar av att hantera avancerade matematiska formler. Allt från integraler och summor till rapportmallar kan enkelt genereras i texten genom realtidsförslag från redigeraren.

Enligt fallstudien*, se kap 2.3*, kan SmartNotes generera symboler och matematiska formler med färre steg än jämförda textredigerare. Färre steg resulterar i en mer användarvänlig applikation som låter användaren effektivisera sitt skrivande och därav ger en bättre upplevelse. I SmartNotes krävs det tre steg för att infoga specialtecken, se bild 3.1, 3.2 samt 3.3.

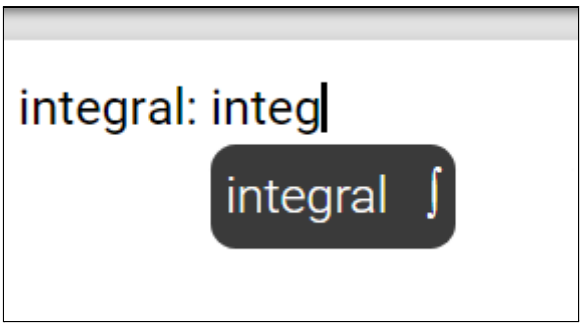

Bild 3.1. Användaren skriver nyckelord och förslag presenteras.

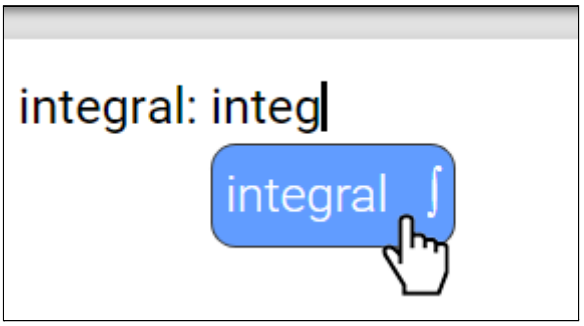

Bild 3.2. Användaren trycker på förslaget som då markeras.

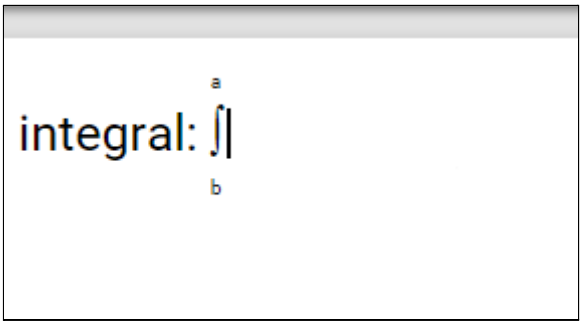

Bild 3.3. Tecknet som användaren har valts infogas i texten

Dessutom har SmartNotes stöd för att hantera speech-to-text för att ge stöd för dels bekvämare redigering samt för de människor som är blinda eller lider av någon annan funktionsnedsättning som försvårar användningen av en textredigerare.

SmartNotes gör det lätt att exportera dokument till applikationens egna format .note och stödjer omvandling till både PDF och LaTeX. SmartNotes kan spara ner det skrivna dokumentet till datorn eller direkt spara det i Google Drive.Vidare kan även dokumentet importeras från användarens Google Drive för att fortsätta laborera och redigera texten i SmartNotes, en flexibel lösning.

4 Analys och diskussion

I detta kapitel presenteras en analys där resultatet diskuteras i relation till målsättningen.

4.2 Diskussion

Vi valde att driftsätta applikationen i AWS för att viss erfarenhet av AWS redan fanns inom teamet. Frontend driftsattes i AWS Amplify för att det har inbyggd integration för kontinuerlig driftsättning. Backend driftsattes i en Linuxmiljö i AWS EC2. En Windowsmiljö sattes först upp, men gratisversionen av EC2 ger en virtuell miljö med en GB ram, vilket ledde problem. Linuxmiljön visade sig vara mindre resurskrävande och skapade inte samma problem med minnesbegränsningar.

För att kunna nyttja all funktionalitet i applikationen behöver den användas via en dator och webbläsaren Google Chrome. Närmare är det Speech to text som, i och med att det använder ett API från Google Chrome, har begränsats till den webbläsaren.

Google Documents komplexa, förhållandevis långa, tillvägagångssätt för infogning av specialtecken och matematiska funktioner, *se kap. 2.2*. Det resulterar till att du kan behöva scrolla för att hitta tecknet du söker, eventuellt scrolla förbi det och bli frustrerad över att “tecknet inte finns”.

I SmartNotes behövs endast två styckna knapptryck för att få fram specialtecken du söker, alternativt endast ett knapptryck om speech to text funktionen används. Startsträckan i SmartNotes är avsevärt mycket kortare då tillvägagångssättet är självlärande, du skriver in ordet eller får upp liknande tecken som förslag där du kan trycka utefter det du valt att skriva in, och sen trycka på tecknet. I Google documents så måste du först hitta bland de olika menyerna och veta hur du ska orientera dig för att hitta ditt tecken du söker.

5 Slutsatser

Det som tagits fram är en webbapplikation som med enkelhet kan, i förhållande till populära marknadsalternativ, generera och hantera specialtecken med med simplicitet, *se kap. 2.2*. Applikationen består av en textredigerare byggd från grunden med funktionalitet för matematiska uttryck, tal-till-text, samt möjlighet att exportera dokument till PDF, LaTex och det egna filformatet .note.

Vid vidare utveckling bör de funktionella avgränsningarna implementeras, så som möjligheten att beräkna matematiska uttryck direkt i textredigeraren och simultant redigera dokument mellan flera användare.

Källförteckning

[1] U.S. Dept. of Health and Human Services. “The Research-Based Web Design & Usability Guidelines, Enlarged/Expanded edition”. Washington: U.S. Government Printing Office, 2006.

[2] HI1036 Mjukvarukonstruktion, projektkurs 7,5 hp, https://www.kth.se/student/kurser/kurs/kursplan/HI1036-20201.pdf?lang=sv, läst 2020-08-05.

[3] MathType, “MathType for Google Docs”, https://docs.wiris.com/en/mathtype/apps/mathtype-google, läst 2020-03-07

[4] G Suite Marketplace, “MathType, https://gsuite.google.com/marketplace/app/mathtype/742924286153, läst 2020-03-08

[5] Google. “Insert Special Characters”. https://support.google.com/docs/answer/3371015?co=GENIE.Platform%3DDesktop&hl=en, läst 2020-03-05

[6] Microsoft. “Insert a symbol”. https://support.office.com/en-us/article/insert-a-symbol-09b3d8e6-cd92-423a-9f5e-7f813e7e4b9e, läst 2020-03-05

[7] Yener, Murat & Dundar, Onur, 2017. Continuous Integration. In Expert Android Studio. Indianapolis, Indiana: John Wiley & Sons, Inc., s. 281–307.

[8] Meyer, M., 2014. Continuous Integration and Its Tools. IEEE Software, 31(3), s.14–16.

[9] Das, Prerana & Acharjee, Kakali & Das, Pranab & Prasad, Vijay. (2015). VOICE RECOGNITION SYSTEM: SPEECH-TO-TEXT. Journal of Applied and Fundamental Sciences. 1. 2395-5562.

[10] CircleCI, “How CircleCI Works”, https://circleci.com/product/#how-it-works, läst 2020-03-05.

[11] Paul Ross, 1995 .“Revival: The Handbook of Software for Engineers and Scientists”.

[12] The Latex Project, An Introduction To LaTeX, https://www.latex-project.org/about/, updaterad 2016-07-28, läst 2020-02-03

[13] React. ”A JavaScript Library for building user interfaces”, https://reactjs.org/, uppdaterad 2011-07-11, läst 2020-02-08

[14] Geeks For geeks. Difference between Typescript and JavaScript, Bishai Kumar Dubey, https://www.geeksforgeeks.org/difference-between-typescript-and-javascript/, läst 2020-03-03

[15] Pusher, “Channels Overview”, https://pusher.com/docs/channels, läst 2020-02-03

[16] Pusher, “Public Channels”, https://pusher.com/docs/channels/using_channels/public-channels, läst 2020-02-03

[17] Pusher, “Private Channels”, https://pusher.com/docs/channels/using_channels/private-channels, läst 2020-02-03

[18] Pusher, “Presence Channels”, https://pusher.com/docs/channels/using_channels/presence-channels, läst 2020-02-03

[19] AWS, “AWS Ampligy”, https://aws.amazon.com/amplify/, läst 2020-02-03

[20] AWS, “AWS EC2”, https://aws.amazon.com/ec2/, läst 2020-03-02

Bilagor

Bilaga 1 - Konfiguration av Circle CI

```yaml
version: 2.1
# Use a package of configuration called an orb.

    # Run the welcome/run job in its own container
jobs:
  build:
    working_directory: ~/redux-async
    docker:
      - image: circleci/node:8
    steps:
      - checkout
      - restore_cache:
          key: npm-cache-v1-{{ checksum "package-lock.json" }}
      - run:
          name: Install Dependencies
          command: npm ci
      - save_cache:
          key: npm-cache-v1-{{ checksum "package-lock.json" }}
          paths:
            - /home/circleci/.npm
      - run:
          name: Run Tests
          command: npm test
      - store_test_results:
          path: test-results
```