Providing Personalised Recommendations of Critical Incident Narratives in a Cross-platform Mobile Application

Tim Wenzel[1], Doris Fetscher[2], Wolfgang Golubski[1], Susanne Klein[2] and Rainer Wasinger[1][a]
[1]*Faculty of Physical Engineering/Computer Sciences, University of Applied Sciences, Zwickau, Germany*
[2]*Faculty of Applied Languages and Intercultural Communication, University of Applied Sciences, Zwickau, Germany*

Keywords: Personalisation, User Modelling, Mobile Applications, HCI, Prototyping, Critical Incident Narratives.

Abstract: This work describes the design and implementation of a cross-platform mobile application that has been created to provide users with personalised recommendations of Critical Incident (CI) narratives. CIs provide brief descriptions of situations in which misunderstandings arise as a result of the cultural differences of the interacting parties. They are useful for increasing intercultural awareness. This paper describes the design and implementation of the mobile application called 'Nils2Go'. The main focus is the personalisation strategy that is employed in the recommendation of CIs, and the identification and use of user demographic characteristics that can be utilised to further personalise a retrieved list of CI narratives.

1 INTRODUCTION

Various studies show that the use of mobile applications often drops sharply within the first few weeks and months after installation. For example, (Yan and Chen, 2011) show how the average usage drops by about 50% over a period of three months. Although also dependant on the type of application (e.g. communication applications are usually used for significantly longer than games that are used exclusively for entertainment (Li et al., 2020)), 'user experience' has been shown to influence long-term application usage (Kujala et al., 2011).

This paper focuses on the question "How can CI recommendations in a mobile application be personalised?". This is important because personalisation has been shown to lead to an improvement in user experience in the past. To achieve this in the context of a mobile application, a user model is created, and personalisation in the form of a recommender system is carried out to present users with a list of relevant CIs.

This paper starts with a definition of Critical Incidents. Following this, Section 2 outlines the background concepts of personalisation, user modelling, and recommender systems, and also provides an outline of the Network Intercultural Learning and Sensitivity (NILS) website. In Section 3, the low-fidelity prototype and the implemented mobile application called Nils2Go[1] are presented, as too the personalisation strategy and technical implementation of the application. Finally, in Section 4, our conclusions and directions for future work are provided.

1.1 Critical Incidents

Critical Incidents (CIs) are brief descriptions of situations in which a misunderstanding, problem, or conflict arises as a result of the cultural differences of the interacting parties (Apedaile and Schill, 2008). CIs form an important tool for increasing our awareness and understanding of human attitudes, expectations, behaviours, and interactions. Some examples of CIs are provided in Figures 1, 2B, and 3.

In addition to the narrative, different metadata about the CI is also stored. For example, it can be relevant where the information came from, who recorded it, and who the actors are. A reflection (i.e. a retrospective view of the past situation) by the author from today's perspective, is also often included in a CI.

The mobile application described in Section 3 incorporates both the CI narrative and its associated metadata into the presentation of CIs to the user. This metadata is of decisive importance in the personalisation of the CIs that are returned to the user and is described in more detail in Section 2.4.1.

[a] https://orcid.org/0000-0003-1069-5814

[1] Nils2Go App: Available at https://play.google.com and https://apps.apple.com.

Wenzel, T., Fetscher, D., Golubski, W., Klein, S. and Wasinger, R.
Providing Personalised Recommendations of Critical Incident Narratives in a Cross-platform Mobile Application.
DOI: 10.5220/0011528400003323
In *Proceedings of the 6th International Conference on Computer-Human Interaction Research and Applications (CHIRA 2022)*, pages 137-144
ISBN: 978-989-758-609-5; ISSN: 2184-3244

2 BACKGROUND

In this section, the concepts of personalisation, user modelling, and recommender systems are outlined, as too the NILS website. These concepts form the basis on which this work is then built upon.

2.1 Personalisation

There are various approaches to increasing the user experience in the area of mobile applications. One of these approaches is the personalisation of the software used. In this context, personalisation involves assigning relevant characteristics to individual users. Based on these characteristics, different users will have different experiences when using the software. The experiences are adapted to the preferences and needs of the user. The work in this paper focuses on how personalisation can be used in the presentation of Critical Incident narratives.

A sub-area of the personalisation of software is represented by so-called recommendation systems. These systems suggest specific information to the user based on matches to his or her personal interests. In general, personalisation deals with the discrepancy between general-purpose applications, which achieve the highest possible benefit in a broad field with the least possible development effort, and specialised applications, which should fulfil an individual-specific benefit (Kuo, 2013).

2.1.1 Advantages and Disadvantages of Personalisation

The development of personalised applications has a number of advantages. One clear advantage, especially for software development companies, is driven by finance. In (Gavril and Ionescu, 2017), the authors demonstrate that personalised applications generate 16.5% to 24% higher revenues. Another advantage that concerns the end user is the increase in user satisfaction. In (Liang et al., 2006), the authors show that personalisation, especially in the form of recommendation systems, has a positive effect on user satisfaction. The authors in (Tong et al., 2012) also demonstrate the positive influence that personalisation has on user satisfaction in software systems. Another positive aspect is that personalised applications often stand out from non-personalised applications (Arora et al., 2008).

In addition to the advantages mentioned, there are also various disadvantages. The disadvantages include the higher development costs and the increased complexity of the development of an application, but these are often offset by higher revenues (Blechschmidt et al., 2005). Another possible disadvantage, especially for the end user, is data privacy, as personal data is often collected and used for user modelling (Blechschmidt et al., 2005). It is however often the case that the positive aspects outweigh the negative ones. In the Nils2Go application described in this paper, user data that is collected never leaves the device.

2.2 User Modelling

Whereas personalisation is the process of providing individually relevant and interesting information for individual users of an application, user modelling refers to the construction of a User Model (UM) that incorporates user-specific data. This model is generated and managed by means of corresponding software components. The collected and managed data can refer to the demographic characteristics of the user being modelled. Furthermore, contextual data of the user can also be relevant. The resulting model reflects assumptions of the system regarding the modelled user and can often be retrieved in the form of a user profile. Depending on the application, such a model can also be designed for reuse in other systems (Kay, 1998).

Examples of user model data relevant to the Nils2Go application that is described in this work include demographic data (e.g. age, gender, countries of interest, spoken languages, and country of residence), general pre-defined topics of interest that the user can select from (e.g. education, food, family, holidays, and travel), and specific interests provided in the form of user-defined keywords. This data is shown in Figure 5D.

In (Vu and Proctor, 2011; Hothi and Hall, 1998), the following types of User Model are outlined:

- **Static UMs:** This is the simplest type of UM. Once the necessary data and information has been provided, the UM remains unchanged for entirety of the application's remaining use. Changes in user interests and preferences are not registered or updated in the model.
- **Dynamic UMs:** Dynamic user models can accommodate for changes in a user's interests and preferences. In this way, there is always an up-to-date model of the user. Furthermore, a user's interactions with the application can also flow into the model. Depending on the use-case, it is also possible to manually adjust the model.
- **Stereotype-based UMs:** Stereotype-based UMs enable a much more anonymous way of generating and maintaining user models. Here, the interests and characteristics of the individual user are

not part of the modelling. Rather, users are classified into common stereotypes. General information about these groups is then used to generate preferences for the user. The basis for these assumptions are primarily driven by statistics that have been compiled for the respective groups of people.

- **Highly Adaptive UMs:** Highly adaptive UMs realise the counterpart to stereotype-based modelling. Here, great amounts of information about a specific user are collected and generated, such that almost all anonymity of the user is lost. One advantage of this UM is that user profiles are extremely specific and accurate. As a result, corresponding applications can be particularly well adapted to the current user and the user receives the best possible user experience.
- **Hybrid UMs:** Hybrid UMs attempt to combine the advantages of both static and dynamic UMs, such that a balance is found between good performance with low memory requirements and good personalisation with detailed user models. Often, as is the case in (Billsus and Pazzani, 2000), the UM consists of sub-models that take both the current point in time as well as a longer period of time into account. This allows information about a user to be collected in order to first generate a static user model, while also allowing for dynamic changes in the interests over a longer period of time.

As described later in Section 3.4 the mobile application in this work incorporates a hybrid UM approach.

2.3 Recommender Systems

In (Melville and Sindhwani, 2010), a Recommender System is defined as a software system that provides relevant and meaningful recommendations or suggestions of different products or information items to a number of users who may be interested in them. Recommender systems usually make use of either collaborative filtering, contextual filtering, or a combination of both. For the purpose of brevity, the reader is directed to (Aggarwal, 2016) for further details on recommender systems. The mobile application described in this work incorporates a content-based filtering approach as is described in Section 3.4.

2.4 NILS: Network Intercultural Learning and Sensitivity

Network Intercultural Learning and Sensitivity (NILS) is a research project of the University of Applied Sciences Zwickau. The work focuses on a scientifically innovative preparation of critical incident narratives (Fetscher and Klein, 2020). In addition to gaining new insights into intercultural communication, the aim of the project is to sensitise users by helping them to question cultural stereotypes. It is assumed that this goal can be achieved through the interactive browsing of CIs (Fetscher and Klein, 2020).

The NILS website[2] has been primarily designed for research, teaching, and learning purposes. It allows researchers (as well as other registered users) to enter CI narratives into the system. It also allows users of the website to retrieve CIs and filter the CIs based on keyword searches. There are currently around 200 CIs available through the website.

Data entry and retrieval of CI narratives is accessible via the project's website, with the work described in this paper providing the missing link to further view CIs in a personalised manner via a cross-platform mobile application that has been developed for Android and iOS smartphones and published on the platforms' respective App store fronts. An example CI narrative from the website is shown below in Figure 1, while Figures 2B and 3 in Section 3 show further example narratives available through the mobile application.

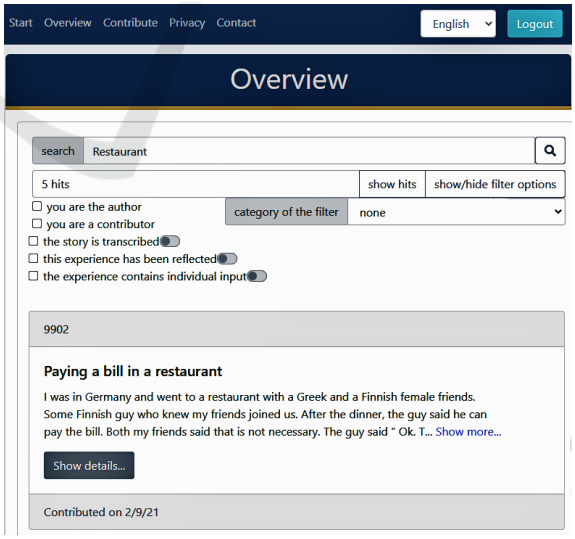

Figure 1: NILS website illustrating a search for relevant CIs based on the keyword "Restaurant".

[2]NILS Website: https://nils.fh-zwickau.de/

2.4.1 Critical Incident Metadata

Each CI has up to 105 attributes associated with it. These attributes can be grouped into the categories: textStory, origin, medium, media, hotspots, contact domains, communication domains, reflection, author, actors, and location. Each of these categories has multiple attributes associated with it, e.g. a textStory includes the narrative (labelled as textStory.story in the dataset) as well as attributes depicting whether it has been transcribed, the language it is written in, and details about the author.

Some of the more important attributes that are collected for each CI include an *ID*, the *title* and the actual *story text*, a *timestamp* of when the data was created, the *origin* (i.e. whether the story is a personal experience, an observation, retelling, or hearsay), the *language* in which the content is recorded, the *kind* of data (i.e. whether it is primary data or secondary/edited data), the narrative *perspective* (i.e. first or third person), the *location* of the event, the *actors* to whom the narrative refers, and personal data about the *author*. Authors also have the ability to reflect on their experience by including a *reflection* at a later time. Authors can also provide *hotspots* in the form of free-text keywords to describe features of the CI (e.g. humorous, historical).

In general, the CIs can be written in any language. In addition to German, the current set of CIs contain some narratives in languages such as English, French, Spanish, and Russian. The actors in the set of CIs come from various European and non-European countries and belong to different age groups.

3 Nils2Go MOBILE APPLICATION

3.1 Purpose and Functionality

The mobile application described in this paper has two main purposes. Firstly, it should allow for the clear presentation of CI narratives included in the NILS dataset. Both the actual CI story as well as relevant metadata (see Section 2.4.1) should be made available to the user. Secondly, the mobile application should offer the possibility to personalise all existing CI narratives within the given framework. In this way, relevant elements classified as interesting for the user are selected from the total amount of data (though still allowing the user to access other CIs via a simple toggle button as shown in Figure 5A).

In contrast to the website, the mobile application has been kept deliberately simple and lightweight. It does not provide the ability to add new CIs, and neither does it allow users to create an account from the mobile device (which would be required to add new CIs). Instead, it has been designed to present CIs quickly and efficiently, and in a manner that is personalised to a given user's profile.

3.2 Low-fidelity Prototype

The mobile application was developed using a User-Centred Design (UCD) approach (Norman, 2013). This process was combined with a semester-long course at the University of Applied Sciences Zwickau[3] that focused on the creation of mobile application prototypes for the search and retrieval of CIs. In particular, the course focused on HCI principles in which 13 groups of students designed low-fidelity prototypes in Balsamiq[4] and iteratively tested their designs, first in a formative and later in a summative manner with 3-5 participants each time. The evaluation technique used for this process was the think-aloud technique (Shneiderman, 2016). These low-fidelity prototypes were used as the starting point for the design of the current mobile application, which then additionally incorporated the concept of personalisation as shown in Figure 2, with the ability to select thematic interests (A) and to browse a list of personalised CIs (B).

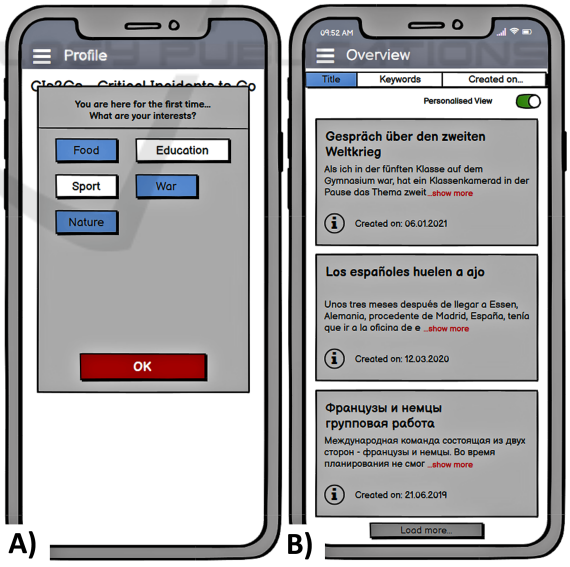

Figure 2: Low-fidelity prototype of the mobile application showing a user's thematic interests (A) and a personalised list of CIs (B).

[3]WHZ, URL: https://www.fh-zwickau.de/
[4]Balsamiq, URL: https://balsamiq.com/wireframes/

3.3 High-fidelity Mobile Application

The mobile application has two main functions. The first is to provide access to the list of CIs. These CIs are packaged locally with the mobile application on compilation. This means that there is no reliance on the Internet or the NILS website to access the CIs. The set of CIs included with the mobile application are however only a subset of the total available to registered users of the NILS website. Updates to the included list of CIs is accomplished via application updates to the respective App store fronts. Figures 3A, 3B, and 3C show the *overview* list in which CIs are arranged vertically in an abbreviated form. This means that for each CI, the title and the first three lines of the actual story are visible. Furthermore, the creation date for each narrative, as well as an information-button that shows the related metadata is provided. To see the complete story, the user simply selects the 'show more' button. Feedback is also provided (when in the personalised mode) to indicate if the match is due to 'Interest Theme', 'Keyword', or 'Demographic characteristic' (Figures 3A, 3B, and 3C).

In addition to the CI narrative, the user also has access to the *metadata* associated with each CI (Figure 3D), which the user can swipe through via a right-to-left swipe gesture (in this case to see details on the CI's Hotspots - Figure 3D top-left) and via a left-to-right swipe gesture (to see details on the CI's Author - Figure 3D top-right). In this view, all of the CI's additional metadata is shown. This information is presented separately to the main story so that it does not impact on the readability of the CI narratives in the overview page.

Also visible in Figure 4 is the ability to view favourited CIs and to sort CIs, both alphabetically, and via the manner in which CIs were selected as being relevant, i.e. based on the 'Field of interest', 'Demographic Features', or 'Keywords'.

The second function is related to the personalisation of the retrieved list of CIs. The information contained in the *user profile* is decisive for the personalisation of the application and contains important data collected about the user.

Figure 5B shows a multi-choice set of topics that is provided to the user during their first use of the application. This allows the application to determine the user's thematic interests for the purpose of personalisation. The user also has the ability to skip this process and configure their profile at a later time via the application's side-menu. Also visible in Figure 5A is the 'Personalised View' toggle-switch that allows a user to see the CIs that are most relevant to them.

The user's profile page is accessed via the side-menu and is shown in Figure 5D. This provides a representation of the user model, which is statically generated at the beginning and - if necessary - adapted over the application's useful life (see Section 2.2). The sections on the user profile page are divided into demographic characteristics and interests. The demographic characteristics list personal information about the user including their age, gender, countries that they affiliate with, spoken languages, and their current location of residence. The ability to enter this data is mainly provided to the user via pre-defined drop-down lists that limit the amount of free-form text that the application needs to interpret. The section on interests provides predefined topics that the user selected on first use of the application, as well as user-defined keywords that can be provided in free-form. At the bottom of the screen is also a button that allows the user to delete all of the information that has been gathered on him or her. If the user selects this button, their intention to delete their profile must be confirmed and the user model is subsequently and irrevocably removed from the application.

3.4 Personalisation in the Application

Section 2.2 outlined static, dynamic, stereotype-based, highly customisable, and hybrid approaches to user modelling. For the Nils2Go mobile application described in this paper, a hybrid UM approach was used. The user model is generated during the first use of the application, similar to a static UM approach. The user can then 'manually' adjust their interests and demographic characteristics, which makes it possible to update the UM without complex dynamic adjustment functionalities. Users can voluntarily further customise their UM with demographic data like age, gender, and spoken languages.

Users can also decide not to provide such demographic data, but the list of returned CIs will then be much less precise as the personalisation strategy will not incorporate CI metadata based on the actors and authors with similar demographic attributes to the user. If the user does initially decline to provide demographic data, they can still decide to add it later via the application's side-menu.

The recommender system used in this application uses a content-based filtering approach. In particular, recommendations are implemented based on rules specified in the software. These rules use the Levenshtein string distance algorithm (Navarro, 2001) to determine similarities between the user's profile data (Table 1, left) and the CIs (Table 1, right).

Using the data in Table 1 row 1 to illustrate, one such rule finds similarities between a user's thematic

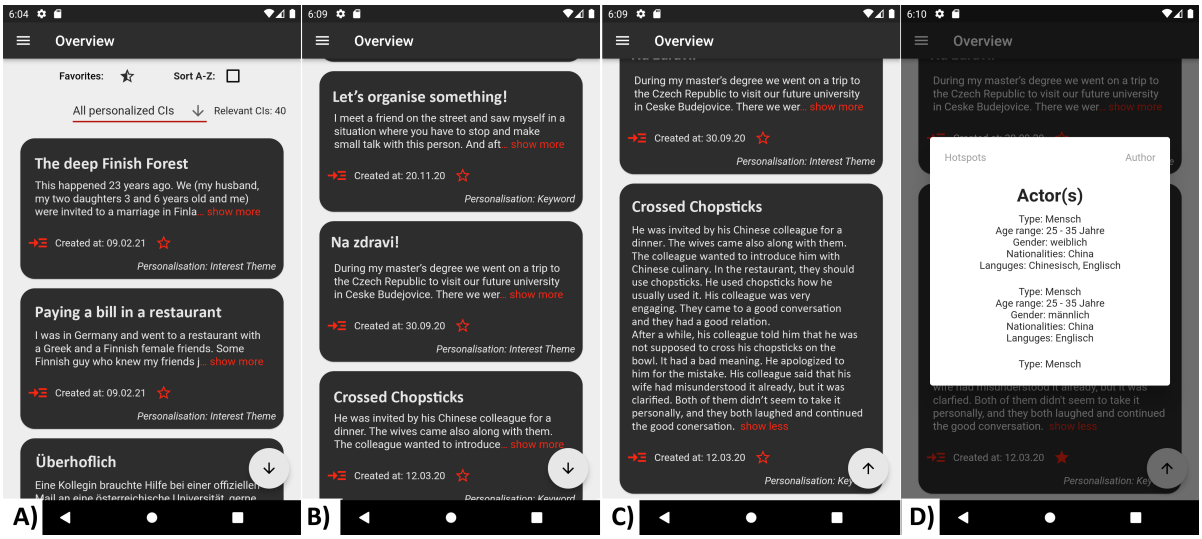

Figure 3: The mobile application for Android, showing the user's personalised list of CIs (A, B, C), and some of the CI metadata (D).

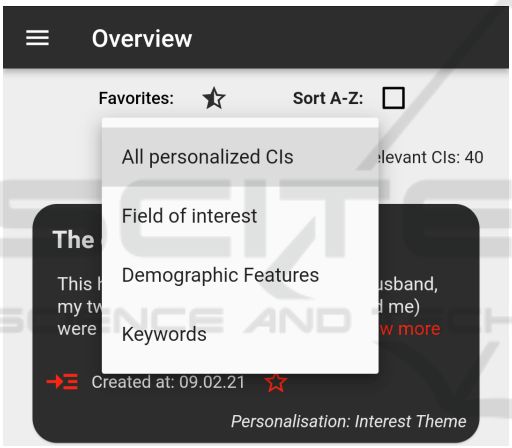

Figure 4: Methods of sorting CIs, including both alphabetic (A-Z) and personalised options.

topics of interest and the CI text story and reflection text. Similarly, if a user provides details on the language(s) that they speak, this will be used to help determine relevant CIs based on the language spoken by the actors, the author, and the language that the CI has been written in (row 5 in the table).

From a technical perspective, the recommendation of relevant CIs is based on the Levenshtein string distance algorithm. This is well suited to applications in which the objective is to find matches for short strings (e.g. our pre-defined themes or user-defined keywords) in longer texts (i.e. our dataset of Critical Incidents). In this mobile application, Levenshtein distance thresholds are used to adjust the relevance of the returned CIs, i.e. if words are less than or equal to the Levenshtein distance, the CI is more likely to be deemed relevant to the user. From a code perspective,

Table 1: Recommender system data. Demographic data is marked with an asterisk (*).

User Profile	Critical Incidents (CI) + Metadata
Thematic Interests	Text story, Reflection
Individual Keywords	Title, Text story, Reflection
Age*	Age (Actors, Author)
Gender*	Gender (Actors, Author)
Languages*	Language (Actors, Author), Language (CI)
Countries*	Nationality (Actors, Author), Country (CI)
Place / Region*	Place (CI), Region (CI)

this is represented by the following variable:

```
const MAX_LEVENSHTEIN_DISTANCE_TO_MATCH = 3;
```

In addition to the Levenshtein distance threshold (which can also be adjusted by the user as shown in Figure 5C), CIs are also determined to be relevant based on the user's demographic data (when available). As outlined in (Beel et al., 2013), the use of demographic data can have a significant impact on the success of the recommender system. In our mobile application, it is a combination of several matching demographic characteristics that is required for a CI to be marked as being relevant based on the user's demographic data (Figure 5C).

3.5 Technical Implementation

In this section, we outline the framework and technologies that were used to build the cross-platform mobile application.

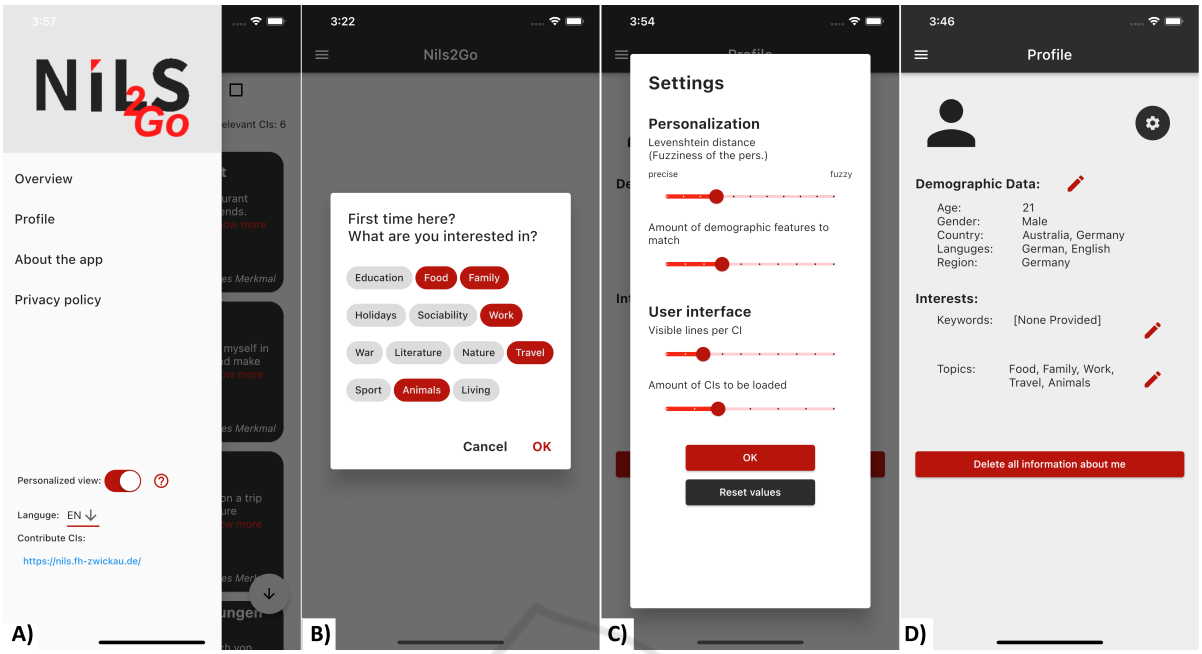

Figure 5: The mobile application for Apple iOS, showing the side-menu (A), the user's thematic interests (B), personalisation settings (C), and the user profile (D).

For the technical implementation of the mobile application, the open source framework Flutter[5] was used in conjunction with the programming language Dart[6]. Flutter is a portable UI toolkit developed by Google for the development of natively-compiled applications for mobile platforms (iOS, Android) as well as desktop (MacOS, Windows, Linux) and web (Web Apps) (Flutter, 2022). Most importantly, this means that the framework offers the ability to develop applications once and then make them available natively on different platforms.

Cross-platform development technologies like Flutter and Dart allow developers to reuse the same code across multiple platforms. They also provide access to native features of the smartphone, including local storage, which is relevant for this mobile application.

The Nils2Go application is based on Flutter v3, which was released in May 2022. According to a 2022 survey by Statista of over 31,743 software developers (Vailshery, 2022), the use of the Flutter framework by software developers increased to 42% between 2019 and 2021 (an increase of 12%). Flutter is thus now already far ahead of competing frameworks such as React Native (used by 38% of software developers in the survey), Cordova (used by 16%), Ionic (used by 16%) and Xamarin (used by 11%). In comparison to web-based cross-development platforms like React Native, Flutter uses its own high-performance rendering engine rather than web technology. The Flutter engine is written in C/C++ and applications written in the Dart programming language are compiled into native code Ahead-of-Time (AoT) for both iOS and Android[7]. This focus on performance means that Flutter applications can run faster than the competing cross-platform technologies.

A comparison of Flutter applications written in Dart compared to native Android applications written in Kotlin and iOS applications written in Swift shows that there is still a performance penalty when using Flutter (Olsson, 2020), but the advantage of having a cross-platform application outweighs the disadvantages for the purposes of this work. In (Olsson, 2020), it is further stated that from their survey of 39 people from the IT industry, 74% of end users could not detect the difference in look and feel between a Flutter and a native mobile application, which provides even more incentive to use Flutter/Dart.

4 CONCLUSIONS

This work described the design and implementation of the Nils2Go cross-platform mobile application that

[5]Flutter, URL: https://flutter.dev/

[6] Dart, URL: https://dart.dev/

[7]Flutter FAQ, URL: https://flutter.dev/docs/resources/faq

has been created to provide users with personalised recommendations of CI narratives. The focus was on the personalisation strategy which was demonstrated using a hybrid user model and a content-based recommender system that incorporated the Levenshtein distance as well as demographic user data (when available). Future work will now be to conduct a user-study with the mobile application to see if the employed personalisation strategy does in fact lead to an increase in user experience for its users as has been shown by other past research.

ACKNOWLEDGEMENTS

A special thank you goes to the PTI06970 HCI students (Alec Sichenender, Jonas Langner, Justin Ntzold, Konrad Platz) and PTI06640 Mobile Applications students (Kenny Musterer, Felix Schober, Dominik Strer, Tommy-Lee Bannert) at the Westschsische Hochschule Zwickau, who worked on earlier prototypes of this application. Special thanks also go to the NILS development team (Tobias Haubold, Wolfgang Grs, Peter Huster, and Dariya Kapinus) and to our graphic designer (David Mantilla), who provided the colour schemes, graphics, and icons.

REFERENCES

Aggarwal, C. C. (2016). *Recommender Systems: The Textbook*. Springer.

Apedaile, S. and Schill, L. (2008). Critical Incidents for Intercultural Communication. Technical report, NorQuest College Intercultural Education Programs.

Arora, N., Dreze, X., Ghose, A., Hess, J. D., Iyengar, R., Jing, B., Joshi, Y., Kumar, V., Lurie, N., Neslin, S., et al. (2008). Putting one-to-one marketing to work: Personalization, customization, and choice. *Marketing Letters*, 19(3):305–321.

Beel, J., Langer, S., Nürnberger, A., and Genzmehr, M. (2013). The Impact of Demographics (Age and Gender) and Other User-Characteristics on Evaluating Recommender Systems. In *International Conference on Theory and Practice of Digital Libraries*, pages 396–400. Springer.

Billsus, D. and Pazzani, M. J. (2000). User Modeling for Adaptive News Access. *User Modeling and User-Adapted Interaction*, 10(2):147–180.

Blechschmidt, T., Wieland, T., Kuhmunch, C., and Mehrmann, L. (2005). Personalization of End User Software on Mobile Devices. In *Second IEEE International Workshop on Mobile Commerce and Services*, pages 130–137. IEEE.

Fetscher, D. and Klein, S. (2020). Wissensplattform fr Critical Incidents und interkulturelle Erfahrungen [Knowledge Platform for Critical Incidents and Intercultural Experiences]. In *Enjeux et dfis du numrique pour l'enseignement universitaire*, pages 79–92. Peter Lang, Berlin.

Flutter (2022). Flutter: Design beautiful apps.

Gavril, L. and Ionescu, S. (2017). Financial Advantages of Software Personalization. *Journal of Information Systems & Operations Management*, 11(1):207–217.

Hothi, J. and Hall, W. (1998). An Evaluation of Adapted Hypermedia Techniques using Static User Modelling. In *Proceedings of the Second Workshop on Adaptive Hypertext and Hypermedia*, pages 45–50.

Kay, J. (1998). *A Scrutable User Modelling Shell for User-Adapted Interaction*. PhD thesis, Basser Department of Computer Science, Faculty of Science, University of Sydney.

Kujala, S., Roto, V., Vnnen, K., Karapanos, E., and Sinnel, A. (2011). UX Curve: A method for evaluating long-term user experience. *Interacting with Computers*, 23:473–483.

Kuo, T. C. (2013). Mass customization and personalization software development: a case study eco-design product service system. *Journal of Intelligent Manufacturing*, 24(5):1019–1031.

Li, T., Zhang, M., Cao, H., Li, Y., Tarkoma, S., and Hui, P. (2020). "What Apps Did You Use?": Understanding the Long-term Evolution of Mobile App Usage. In *Proceedings of the Web Conference 2020*, pages 66–76.

Liang, T.-P., Lai, H.-J., and Ku, Y.-C. (2006). Personalized Content Recommendation and User Satisfaction: Theoretical Synthesis and Empirical Findings. *Journal of Management Information Systems*, 23(3):45–70.

Melville, P. and Sindhwani, V. (2010). Recommender Systems. *Encyclopedia of Machine Learning*, 1:829–838.

Navarro, G. (2001). A Guided Tour to Approximate String Matching. *ACM Computing Surveys*, 33(1):3188.

Norman, D. (2013). *The Design of Everyday Things*. Basic Books.

Olsson, M. (2020). A Comparison of Performance and Looks between Flutter and Native Applications. Bachelor's thesis, Faculty of Computing, Blekinge Institute of Technology, Sweden.

Shneiderman, B. (2016). *Designing the User Interface: Strategies for Effective Human-Computer Interaction*. Pearson.

Tong, C., Wong, S. K.-S., and Lui, K. P.-H. (2012). The Influences of Service Personalization, Customer Satisfaction and Switching Costs on E-Loyalty. *Int. Journal of Economics and Finance*, 4(3):105–114.

Vailshery, L. (2022). Cross-platform mobile frameworks used by software developers worldwide in 2019 to 2021. Technical report, Statista.

Vu, K.-P. L. and Proctor, R. W. (2011). *Handbook of Human Factors in Web design*. CRC Press.

Yan, B. and Chen, G. (2011). AppJoy: personalized mobile application discovery. In *Proceedings of the 9th International Conference on Mobile Systems, Applications, and Services*, pages 113–126.