Software Architecture

Chiel Peters, Omar Pakker, Mary Gouseti, Cindy Berghuizen

October 17, 2013

1 Introduction

1.1 Scope And Summary

This document provides the current architecture of the FlyWithUs application. FlyWithUs is a new start-up that wants to create a single platform for airlines and users to write and read flight reviews. The platform must combine reviews from other sites as well as reviews from within the FlyWithUs application. This document identifies the stakeholders, requirements, design decisions and viewpoints on the current architecture of the FlyWithUs application and contains an overall definition of architecture. Because FlyWithUs does not currently have any technical employees (developers, testers etc.) these stakeholders are currently left out of scope of this document.

1.2 View Overview

The document contains three viewpoints on the architecture:

1. *Functional Decomposition view.* This viewpoint identifies the different users within the system and their functionality. It also shows which modules are responsible for which actions.
2. *Data Flow.* This viewpoint is responsible for showing how data flows through the system.
3. *Concurrency view.* This viewpoint explains the performance and scalability of the system along with the tactics used to achieve those.

1.3 Document Organization

In the first section, the stakeholders are identified together with their concerns. The second section contains multiple viewpoints on the architecture of the FlyWithUs system. In Appendix A the requirements of the stakeholders are stated. The design decisions are given in Appendix B and finally Appendix C contains the domain related analysis.

2 Stakeholder Concerns

In this project there are four different stakeholder :

1. The Initiator
2. AirFrance - KLM
3. The Dutch Government
4. EU Claim

Each of them had different concerns within this project which are stated in the next subsections.

2.1 Initiator

The initiator is the one who started this project. He wants FlyWithUs to be a success and become the number one rating site people will go to. He want to be able to collect data from social media, other rating websites, news, weather and every other source that can be of importance. This data an then be used to provide the airlines with powerfull statistics and can give users a final rating. Users that make use of FlyWithUs have to be able to post reviews and ratings on the website and search for them. What makes FlyWithUs unique is the fact that airlines can get in touch with the users by sending them messages.

2.2 AirFrance - KLM

This stakeholder wants to have a reporting tool. With the tool he has to be able to see what recent reviews have been posted about his airline. Also, AirFrance - KLM wants to see statistics and be able to see what causes a sudden decline or increase in the rating. Furthermore, AirFrance-KLM wants to be able to enter flight information and by doing so influence the weight of review. This means that the weight of a rating has to be less when bad ratings are due to for example environmental issues (bad weather etc.) and have nothing to do with the airline companies services.

2.3 Dutch Government

Privacy is an important issue for the Dutch Government. The server needs to be hosted in the Netherlands so FlyWithUs will be led according to the Dutch Privacy Law. Also, the Dutch Governemnt would like to see the project to be a "Green IT" project.

2.4 EU Claim

EU Claim wants to make certain that the privacy if the user is guaranteed. Furthermore the airlines have to behave on the website and do not mess with the results or bribe the users. Fairness is thus also an important issue to this stakeholder.

3 Viewpoints

This section contains three viewpoints on the architecture. These viewpoints were mainly due to the stakeholders concerns. The privacy view provides EU-Claim and the Dutch government the information related to their main concerns. The data-flow indicates how the system will be able to handle all the data from external and internal sources which is one of the biggest challenges in designing this architecture. The last view shows the functional requirements of the system of all the available users.

3.1 Functional Viewpoint

- Related stakeholders: KLM, Initiator, EU-Claim
- Related Concerns: Users of the system, Available functionality to each user group, Grouping functionalities
- Related design decisions: EU-Claim can see private conversation after invitation, Billing system added, Functionalities of Reporting System.

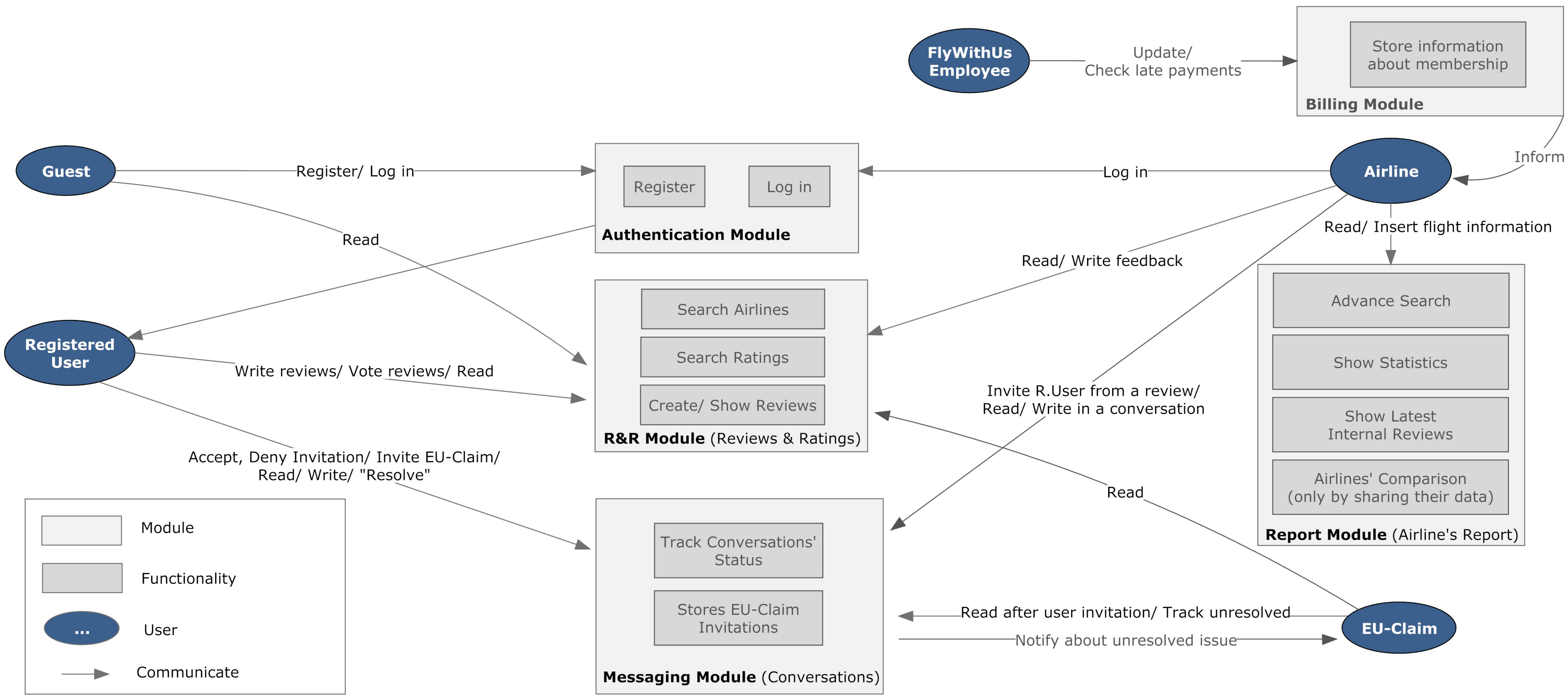

Figure 1: Functional Viewpoint

The functional viewpoint (Fig. 1) illustrates the functionality supported by the system and its connections to the users. The direction of the arrows shows who or what initiates this connection, for example EU-Claim can monitor a conversation (initiated by EU-Claim) but the system notifies EU-Claim about an issue that has been unresolved for a configurable period of time (initiated by Messaging module). The functionalities that are based in the same entities of the system are grouped together to help the reader understand their connections. For example, all the functionalities provided only to airlines are included in a *report* as a result these functionalities are grouped in the Report module. The association between the modules presented in the viewpoint and the functional requirements listed in table **??** is shown in table 1.

Requirement	**System**	**Users**
Show ratings	Review System	Guests, Registered Users
Post/Read/Search/Vote	Review System	Registered Users
Messaging	Messaging System	Registered Users, Airlines, EUC
Reporting	Reporting System	Airlines
Flight Information	Reporting System	Airlines
Statistics	Reporting System	Airlines

Furthermore, the viewpoint also presents the availability of the functionalities to each user group and the communication between these groups through the system. For instance, a guest can register or log in the system and become a registered user, or an airline can invite a user to a private conversation in order to resolve a potential complaint; then the user can accept the invitation and he can also invite EU-Claim to monitor their conversation to ensure the integrity of this process.
Each entity presented in the viewpoint is discussed in the following sections.

3.1.1 Users

Guests. A guest is a user of our system with no account. He can read reviews and search for ratings using the R&R module. Furthermore, a guest can register or log in the Authentication module.

Registered User. A registered user can read, write and vote reviews and search for rating using the R&R module. Additionally, he can communicate with an airline through the Messaging module in private, meaning that other parties (except for EU-Claim and only if it is invited) cannot read this conversation.

Airlines. An airline represents the airlines that have payed membership in FlyWithUs. An airline can monitor its progress checking the Report module. Also an airline can see statistics, ratings and search according to flight information it can provide.
Additionally, as far as customer satisfaction is concerned, an airline can write feedback to a review posted on the R&R module or invite a user to a private conversation through the Messaging module.
Finally, it can be informed about the status of its membership thought the Billing module.

EU-Claim. In order to ensure the integrity of the system, EU-Claim will have separate accounts that will enable her to monitor conversations when she is invited and also notify her of a conversation is unresolved for a configurable period of time.

FlyWithUs Employee. FlyWithUs employee is an employee of FlyWithUs whose responsibility is to update or check the membership status of the airline according to their payments.

3.1.2 Modules

Review & Rating Module. This module is responsible for inserting new internal reviews on the system, vote reviews and provides the search interface to the registered users and guests. This module allows searching according to specific rating fields for example which airline has the best food or the best client support, or according to general ratings.
The R& R module does not allow the users to edit or delete their rating in order to preserve the system's consistency and integrity.

Report Module. This module provides the tools to the airlines to monitor their progress. It shows statistics about the airline progress, the latest reviews on FlyWithUs and also allows the airlines to use advance search, which enables them to insert flight information or other parameters and search through our database.

Messaging Module. This module is responsible for the private conversations between the registered users and the airlines. At first, an airline declares that wishes to contact the author of a review, then the author is invited to a private conversation. If the author accepts it can invite EU-Claim to monitor this conversation through this module. Additionally, this module keeps track of all the resolved and unresolved issues, it can provide statistics and it can also notify EU-Claim if a conversation has been unresolved for a configurable (by EU-Claim) period of time. Only the affiliated parties of a conversation can read this conversation as a result the Messaging module checks the identity of the user before it grants him access.

Billing Module. This module is responsible for keeping track of the membership status of the airlines. It will notify both the FlyWithUs employee and the associated airline if there is a late payment and in case that the airline's membership is not renewed it will revoke the airline's access to the Reporting module.

Authentication Module. This module is responsible for the registration of new users and for signing in existing users and identifying their type.

3.2 Data Flow Viewpoint

- Related stakeholders: KLM, Initiator
- Related Concerns: Data Integrity, Performance, Scalability
- Related design decisions: How is the data split up into multiple databases?; How do we handle large data-sets?; Recalculate or Combine rating?; Incremental or Time-interval update?; Flight-Information

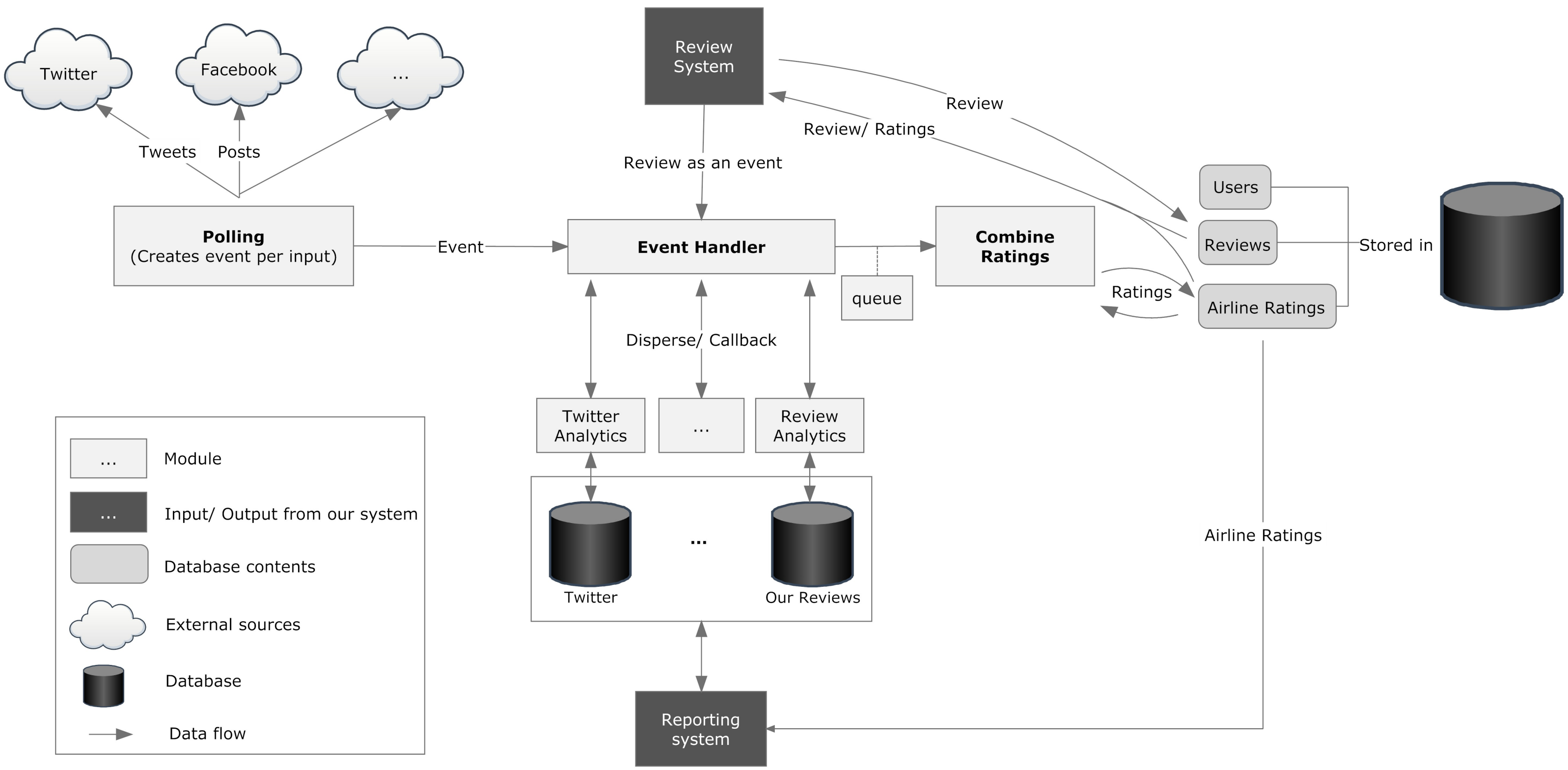

Figure 2: Data Flow Viewpoint

This Data Flow viewpoint 2 illustrates the flow of data within the back-end of the system. In this picture the review and reporting system are shown as black boxes to illustrate that their functionality is of no importance in this picture. In the related design decisions it was argued that combining the rating with incremental updates was best for performance and scalability. Note that in the flow only the back-end of the system is given, because of the importance of handling the large data sets. Each of the modules and arrows in the viewpoint are explained here after.

3.2.1 Polling

The polling module is responsible for obtaining reviews from external data sources. It polls every external source on a given time-frame which is different for each external source. The reviews are not formatted and there original data format from the external source is kept. The rationale behind this decision is that all the reviews from external sources are significantly different of each other. Parsing them into a general format would lead to data loss or lots of undefined fields. The decision is further elaborated in the appendix at 7. The unformatted reviews are then passed on to the event handler as events. An event is defined as a single unformatted review either from external or internal sources.

3.2.2 Event handler

The event handler receives the events from both the polling module (external sources) and the review system (internal source). The responsibility of the event handler is to send these events to the analytics module specific to that data source. The event handler allows for parallel processing of reviews and greatly increased the performance and scalability compared to a more traditional pipeline model. This decision is further elaborated on in the appendix at 2.

3.2.3 Analytics

Each data source has its own analytics module, because the data must be treated differently from external sources as they are of different format. The separate analytics modules have the added benefit of allowing for meta-analysis specific to a given data source. An example could be the amount of followers on twitter. The analytics module analyses the unformatted review and produces a rating on a given scale for a certain or multiple categories (Overall, food, timeliness etc.). The unformatted review together with the analysed ratings are then stored in the analytics database of the data source. This is a requirement by the Initiator and KLM. The decision is further discussed in the appendix B. The ratings are also send of the combine ratings module in order to update the rating of the airline that is related to the review.

3.2.4 Combine Ratings

The combine ratings module is responsible for combining the ratings from the individual reviews into a final rating for an airline. The module receives the ratings from the analytics module and obtains the old rating from the main database. These two are combined in order to form a new rating. The process of iteratively combining ratings instead of recalculating is made for performance reasons as it is much cheaper and efficient to not recalculate the ratings that are already in the final rating of the airline. This decision is discussed in the appendix at 3. The ratings are based on an airline and a specific category (overall, timeliness, food etc.).

3.2.5 Modules

The functionality of each of the modules is discussed in the functional view. Hence in this section only the data flow regarding each of the modules is discussed:

1. *R & R module* A new review is inserted as event into the event handler which can pass it off to the specific analytics module. The decision was made to save the internal reviews twice: once in the main database and once when the they are analysed in the Analytics database. This allows the users to

see the reviews without accessing the analytics database, but increases overhead as the same data is almost saved twice.

2. *Reporting module* The reporting module retrieves information from the analytics database which contain all the analysed data together with the raw reviews. It must also be able to access the main database for the latest final ratings.

3. *Authentication module* The user data for logging purposes is saved in the main database. The data contains sensitive information and is therefore encrypted.

4. *Billing module* In order to make a profit airlines need to pay for their functionality. Therefore the billing module is able to read the current enlisted airlines and write to the database if the airlines have paid or not.

5. *Messaging module* The messages are saved separately in the main database. Because they can contain user sensitive information (e.g. Flight numbers, names) the whole messages are saved in an encrypted form.

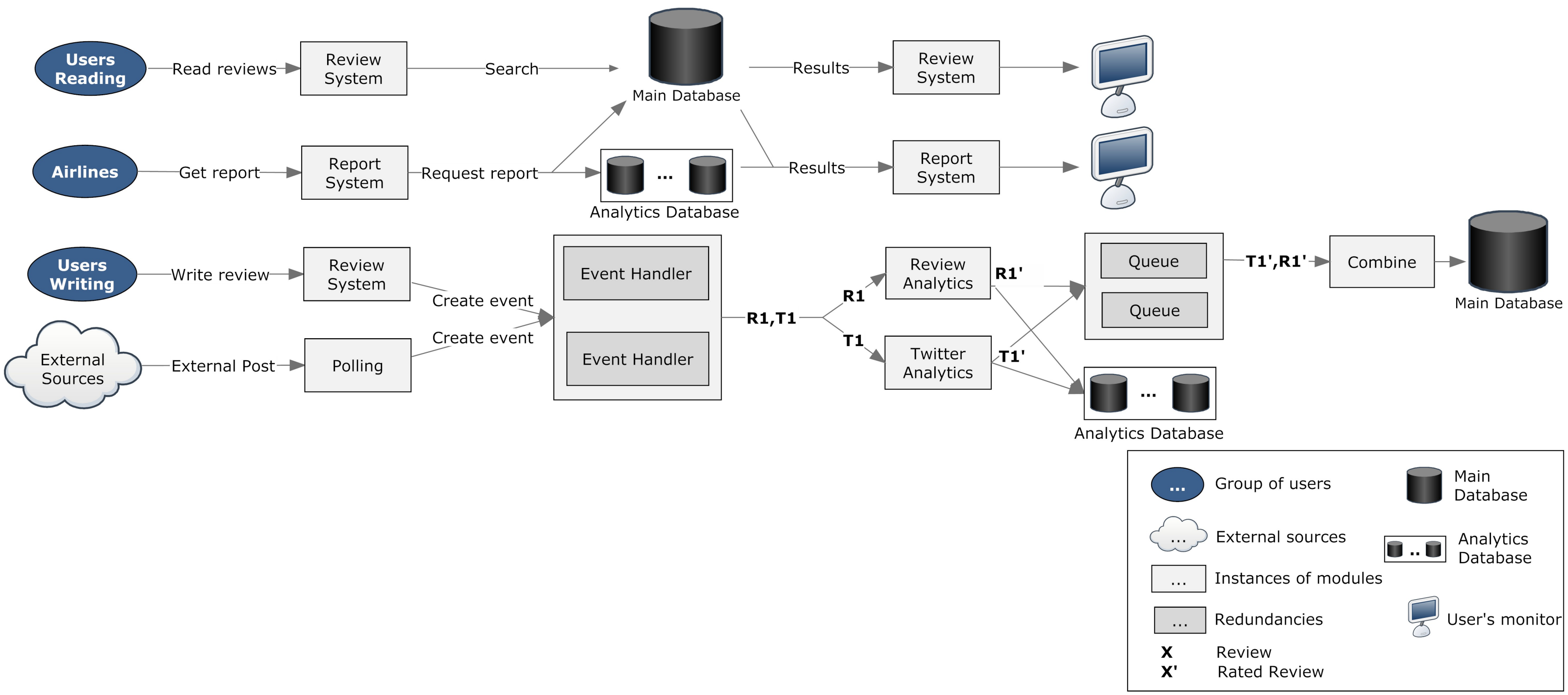

Figure 3: Concurrency Viewpoint

A Requirements

This Appendix describes the functional and non-functional requirements given by the stakeholders in different meetings. The stakeholders ,with the abbreviation used in this appendex, involved with these requirements are: the initiatior (Init), the Dutch government (Dg), EU-Claim (EUC) and AirFrance-KLM (KLM).

Functional Requirements

The functional requirements are given in prioritzation order, the first requirement being the most important.

Requirement	Abbreviation	Related stakeholder
Show the ratings and reviews of all airlines to all users	Show ratings	Init
Post, read, vote and search reviews including ratings	Post/Read/Search	Init
A messaging system where airlines can get in touch with customers	Messaging	Init, KLM
Collect data from external sources like twitter and tripadviser	Data gathering	Init, KLM
Combine the data from external sources to an airline rating	Data combining	Init, KLM
A reporting tool	Reporting	KLM
A transparant and fair way to calculate the airline ratings	Transparancy	EUC
Store the (raw) external data	Save	Init
The system should be able to keep track of the B2B users for offline payment	Billing	Init
Provide flight information to influence the weight of a review	Flight Information	KLM
The ability to see the statistics of other airlines if permission is granted	Statistics	KLM

Non-functional Requirements

The non-functional requirements are given in prioritzation order, the first requirement being the most important.

Requirement	Abbreviation	Related stakeholder
The system should give the data within two seconds	Performance	Init, KLM
The user data should only be available for the user itself and not for the public.	Privacy	Dg, EUC
Scalability is needed for both the users and the amount of reviews.	Scalability	Init
External datasources should be addable without affecting the rest of the system	Adding datasources	Init
The server needs to be hosted in the Netherlands	Hosting	Dg, EUC
The project should be a GreenIT project	GreenIT	Dg

B Formal Design Decisions

This appendix states the design decisions that were taken during the process of designing an architecture. Figure 4 shows the tree of the design decisions that will each be further elaborated on this section.

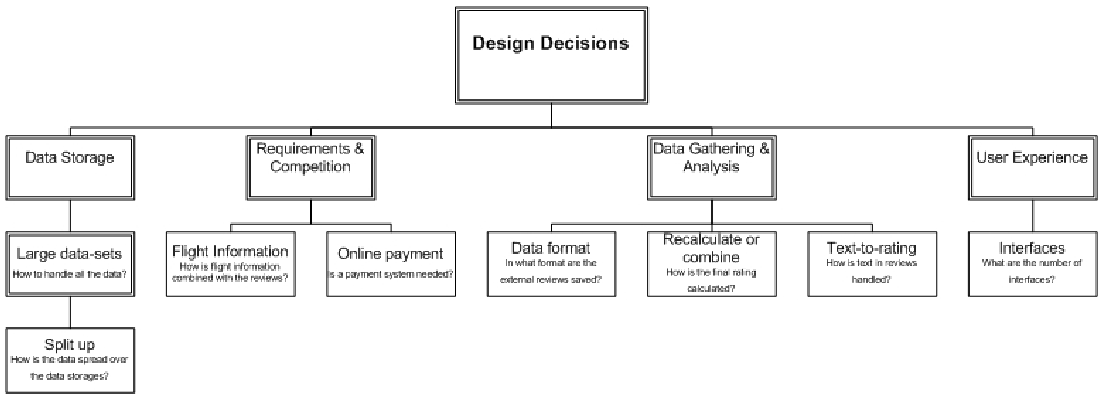

Figure 4: The design decision tree

Design Decision 1: Data format

Issue	The external sources have a unique format related to them. The system needs to be able to combine all these reviews into a final rating of a generalized format. How will we deal with the unique format of the external sources?
Decision	The unique format is kept in before it is analysed. The analysing step converts the unique format into a rating on a category (e.g. food, timeliness) which the system can use and combine.
Status	Decided
Group	Analysis of data;
Assumptions	The data from external sources cannot easily be combined because they all have a significantly different format.
Constraints	None
Positions	Keep the unique format before analysing the data. Parse the unique format from external sources into a general format the analyses module can use.
Argument	The external sources contain different layouts parsing them into a general format would either lead to data loss or lot of undefined fields. However the analyses module does not need to account for different kinds of data which decreases its complexity. The potential loss of data (and maybe meta-data) would however decrease the effectiveness of the analyses module. Therefore the decision was made to keep and store the data in a unique format.
Implications	The storage system needs to be able to handle the data from external sources. The analysis module needs to handle multiple lay-outs of the external sources.
Related decisions	
Related requirements	Data combining, Data Storage

Design Decision 2: Flight Information

Issue	It may occur that a flight is delayed due to external factors outside of the airlines responsibilities (e.g. weather). A bad review may affect the airlines rating which may not be considered a desirable effect. Do we want Flight Information to influence the rating of an airline or shall we not regard it.
Decision	The status is still pending awaiting conversation with the involved stakeholders.
Status	Pending
Group	Analysis of data;
Assumptions	Users will write reviews even if factors are outside of the airlines responsibilities.
Constraints	None
Positions	Cross-match flight information from airlines into the rating algorithm to avoid the influence of those reviews towards the rating Let the airlines search for reviews based on flight information, but not change the rating algorithms. Leave the problem out the scope of the system
Argument	If airlines can influence the rating algorithms than this greatly decreases the transparency of the system. This problem applies to each airline so it does not decrease the fairness of the system. The second option gives airlines the ability to spot the influence of those bad reviews, but not alter their influence. This keeps the system transparent.
Implications	The implications are given after the decision is taken.
Related decisions	Split-up
Related requirements	Flight information, Transparancy

Design Decision 3: How are large data-sets handled?

Issue	Data collected from external sources requires a great amount of storage over time and will grow even bigger over the years. The data also needs to be analyzed and have statistics generated/stored.
Decision	Event type of dispatching.
Status	Decided
Group	Data storage
Assumptions	The number of incoming data sets is that of a big data type.
Constraints	None
Positions	Pipeline model Event type of dispatching
Argument	In a pipeline model each step (Storing, analyzing, Storing statistics + rating) is performed in a sequence. Every review from every external source goes through the same system pipe. This is bad for scalability reasons as more and more reviews would clog up the system. The event type of dispatching can disperse reviews through a channel depending on their source. This creates paralism in the system and is more easily scalable. By collecting the data and sending it to the analyzers, FlyWithUs can generate the rating and score before storing it all. This way there is no need to read and write twice from intermediate storage which improves performance.
Implications	An event handler needs to be developed with clear contracts towards the other components within the system.
Related decisions	How is the data split up into multiple databases?
Related requirements	Performance, Scalability

Design Decision 4: Recalculate or combine rating?

Issue	When receiving new information from external sources or our own website, the rating for the airline company needs to be updated. The rating can be recalculated by using all the historical data (full recalculate) or it is possible to combine only the new information that is not in the old rating and combine them (combine).
Decision	Combine the old rating with the new information (reviews) to update to a new rating
Status	Pending
Group	Data Analysis
Assumptions	It is possible to keep quickly keep count of the new reviews that are not within the current rating.
Constraints	None
Positions	Recalculate the whole rating Combine the old rating and new information
Argument	Recalculating the whole rating is a performance expense operation and leads to a lot of duplicative steps. This is especially true when the number of reviews (external and internal) starts to grow. However in the case of combining, if a cross match needs to occur between flight information and reviews then the flight information needs to be available before the reviews come in. This is because once combined it is included in the final rating and cannot be reverted back.
Implications	If the choice is made for combining than the flight information system needs its information imported before any reviews come in.
Related decisions	Incremental or Time-interval updates; Flight Information
Related requirements	Data combining, Flight Information

Design Decision 5: How is the data split up into multiple databases?

Issue	The system deals with a lot of stored data, either new or already analysed and from a lot of different resources. How many databases gives a good maintainable system and allows for the best performance for all the different database usages?
Decision	Split between data-mining (external and internal reviews statistics) and website data (users, conversations , ratings)
Status	Decided
Group	Data Storage
Assumptions	At the very least there is website data, data-mining data and statistical data.
Constraints	None
Positions	All data is contained within one large database. Split between data-mining (external and internal reviews statistics) and website data (users, conversations , ratings) Split between data-mining (external reviews), statistics on (external) reviews and website data (users, conversations , ratings)
Argument	Keeping all the data in one large database makes it very hard to scale and generally creates performance issues related with the amount of incoming data. The third option leads to a greater number of databases. However the statistics are heavily related to the reviews themselves (data mining) splitting these two up creates fragmented data.
Implications	There is a need for multiple databases that have some relation to each other. These relations need to be clearly defined in order for the system need to completely get out of synch.
Related decisions	How do we handle large data-sets?
Related requirements	Data Storage, Performance, Scalability

Design Decision 6: Text-to-rating

Issue	A lot of reviews contain text or are text-only. How is the text within reviews used in order to form a rating on a given scale?
Decision	The system needs to be able to handle text-only reviews. Therefore a sentiment analysis must be done on the text in order to transform it to a scale
Status	Decided
Group	Analysis of data; Data gathering;
Assumptions	The reviews from external sources contain text or are text-only. This is assumption is based on the observations in social media (e.g. Twitter, Facebook)
Constraints	None
Positions	Drop all the text within reviews and only use available ratings from external sources Perform sentiment analysis to transform text to a rating.
Argument	Dropping all the text within reviews leads to a great loss of information and data. Although it would the most objective solution the loss of data is considered to be too big of a loss. Therefore it was decided to review the text as well.
Implications	Because the system needs to handle with text a sentiment-analysis needs to be done. This requires additional functionality to the system.
Related decisions	Data format
Related requirements	Data Storage

Design Decision 7: Redundancy?

Issue	Single points of failure form a risk in the system. The whole system may become unavailable if these points crash or are otherwise inhibited from doing their work. To keep the system up and running, critical modules need to have some form of backup/redundancy.
Decision	Identify single points of failure and implement redundancy.
Status	Decided
Group	Redundancy
Assumptions	Single points of failure prevent the system from working properly.
Constraints	None
Positions	Do not analyse single points of failure and redundancy. If the system crashes, it crashes.
Argument	By implementing redundancy for single points of failure, the system will be able to switch over to a redundant instance and thus keeping the system running. Without redundancy, every part of the system that uses the module that failed will also be unable to continue.
Implications	By implementing redundancy, more resources are required. This may increase operational cost if a new server is required. The performance of the redundant module may decrease because the module now needs to keep the redundant module in sync.
Related decisions	None
Related requirements	Backup/Redundancy

Design Decision 8: Save the raw data of external sources?

Issue	Several external sources have a limit of how far back you can request data. This means that after a given amount of time (depending on the source) you will be unable to recollect this information. As the system analyses this data and assigns a rating based on an algorithm, it may be possible that improvements or adaptations to the rating algorithm influences the ratings of incoming data and thus of previously analysed data as well.
Decision	To be able to update the ratings when the rating calculation algorithm improves, the raw data of external sources is stored so a recalculation of the ratings can be done at a later date.
Status	Decided
Group	Data storage
Assumptions	External sources limit how far back you can collect data.
Constraints	None
Positions	Generate a rating but do not store the raw data.
Argument	If the algorithm that calculates the ratings is adapted and/or improved, the rating assigned to the same data item may differ. Because of this it would be impossible to recalculate the ratings if the raw data was not saved. If the data is not saved it would mean that either the ratings become a combination of ratings generated by different versions of the algorithm and thus creating an inconsistent rating, or that the previously calculated ratings should be excluded; a rating restart.
Implications	Saving the raw data from external sources requires more storage space.
Related decisions	None
Related requirements	Save

C Domain Knowledge - Design Decision

In this appendix we will present the domain specific problems related to airline reputation management system. These provide a basis for making and finding the design decisions in Appendix B. To divide the workload four separate directions were defined. These directions are highly interconnected and might not appear as separate in the final design. The four directions are:

1. User Experience
2. Competition and Functionality
3. Data storage
4. Data Gathering

A separate section has been dedicated to each of the direction.

C.1 Domain Specific Problems Related to User Experience

C.1.1 Important Questions

The purpose of this section is to ask domain specific question related to user experience. The interface consists of all front-end systems that are directly in contact with the users. In order to get a better insight over the domain a market research was performed with related airline websites. In this research the importance was placed on the users and what they are looking for in a rating site. The questions that are important to the interfaces which affect user experience are:

1. How many and what types of customers are distinguished in the system?
2. What kind of interactions should each customer be able to perform?

C.1.2 Answers

Question	**How many and what types of customers are distinguished in the system?**
Answer	During our meetings with the stakeholders it was made clear that there are two types of customers: 1. Clients: In general, the people who write and read the reviews and ratings. 2. Business Clients: The airline companies. The airline companies want to see a general overview of their companies ratings and reviews with some additional features such as the reporting system (discussed in the next section).
Design Decisions	NaN

Question	What kind of interactions should each customer be able to perform?
Answer	• Clients (Registered Users): 1. They can write (*anonymously*), read and vote the reviews (Review System). 2. They can see and compare the ratings of different airline companies. 3. They can accept the invitation of an airline based on one of their reviews to a private conversation (Messaging System). 4. They can Invite EU claim to a private conversation. • Business Clients (Airline Companies): 1. They can write feedback to a user's review. 2. They can invite a user to a private conversation based on a review in order to resolve his/ her complaint. 3. They can see the current ratings and/ or the latest (bad) reviews posted (Reporting System). 4. They can see statistics about their progress. 5. They can see their progress in comparison with other airline's progress (only if they share their information as well). 6. They can insert flight information and search through our databases to find patterns. The fact that these two types have clearly different functionalities results in two options concerning the system's interface, either two totally separate sites, or a general page whose content will change depending on the signed in user but the lay out will be the same. In addition to these types there are also Guests and EU-Claim: • Guests: Unregistered users who can only search for airlines and read reviews and ratings. • EU-Claim: EU-Claim can monitor a private conversation after a user's invitation. Furthermore, is notified if a conversation is not "resolved" for a specified period of time.
Design Decisions	Separate interfaces for Clients and Business Clients

C.2 Domain Specific Problems Related to Competition and Functionality

C.2.1 Important Questions

The purpose of this section is to find the domain specific questions related to our functionality. To accomplish that, we searched in the internet to learn more about the competition and what they offer in order to adapt the system's architecture to focus on the functionalities that are going to differentiate from them. The questions that help FlyWithUs understand the domain are:

1. Is online payment necessary?

2. How do the feedback and the messaging systems work?
3. How will the reporting system work?
4. How can results of FlyWithUs be protected?
5. How is users' privacy ensured?
6. Which features of the review format are going to ensure users' privacy?

C.2.2 Answers

Question	**Is online payment necessary?**
Answer	It was requested that the services to the airlines are available after payment. The payment can either take place online or offline. In the first case, the stakeholders are going to have an extra cost (fee: a percent of every transaction or predefined + 10$ 25$ every month). On the other hand, since the payment is addressed only to airline companies it is possible to do that offline through contracts, invoices etc. The stakeholders informed us that they want to have an automated billing system.
Design Decisions	Billing system

Question	**How do the feedback and the messaging systems work?**
Answer	The platform of FlyWithUs provides to the airlines the opportunity to respond to a review if they believe it is necessary. (Feedback system) Additionally, an airline will be able to invite a user to a private conversation based on a bad review in order to address an issue or to try to compensate him/ her. The user can choose if he/ she wants to continue and if he/ she wishes to provide his/ her information. It is also significant to decide whether preserving the integrity of the system's messaging service is desirable. To accomplish that, the a User is able to invite EU-Claim to a conversation. Furthermore, it is desirable for such a conversation to have states for example resolved and pending, then a user can change its state from pending to resolved. As a result EU-Claim can be notified by the system if a report has been unresolved for a certain period of time. Additionally, the state of a conversation allows the system to calculate the resolved issues and include that result in the website in order to encourage people to use the FlyWithUs platform.
Design Decisions	NaN

Question	How will the reporting system work?
Answer	The reporting system is a functionality only available to airlines. Each airline will be able to monitor its rating status through a report webpage which will comprise of the following components: • Statistics concerning its progress through graphs and diagrams. • Latest reviews or bad reviews from our site. • Cross-reference flight information with the reviews stored on our system. • Comparison between airlines (only if the airline shares her information as well).
Design Decisions	Flight Information

Question	How can results of FlyWithUs be protected?
Answer	The copyright of the data that will be collected should be further investigated according to the policy of each source. If the legality of harvesting data from the other sources is resolved and even if the data gathered are public, the collection of the data on our system can be protected. This can be achieved because even though the data are public a collection of them is copyrightable. Additionally, we can use a file called robots.txt to prevent data aggregation from specific crawlers. However, the effectiveness of this file lies entirely on the crawlers because they can ignore the file. It appears that there are no architecture relevant decisions on this question.
Design Decisions	NaN

Question	How is users' privacy ensured?
Answer	As it was mentioned in the table: Feedback & Messaging Systems, users' privacy is an important issue. This issue was raised by three of the stakeholders: the Dutch government, EU-Claim and the initiator. A user's data and conversations with airlines are private, this means that these information are available only after user authentication and in the case of the private conversation available only to the airline of interest (maybe to EU-Claim as well if the user requests that). Furthermore the user data will be in an encrypted database.
Design Decisions	Userdata will be encrypted

Question	Which features of the review format are going to ensure users' privacy?
Answer	Since user's privacy is a big issue the review's format should allow anonymity. This is accomplished by omitting the name, username or any other author identifier from the public representation of a review.

Design Decisions	User information will not be shown in the review

C.3 Domain specific problems related to data gathering and analysis

C.3.1 Import questions

The purpose of this section is to explain domain related problems related with this task. The data gathering is responsible for polling external sources and providing the data to the storage system. This implicates two important question which are further elaborated next:

1. What different kind of external resources are we dealing with?
2. How is the data from different external sources combined?
3. How is the combined data used in order to obtain a rating?
4. Is there any filtering process on the external data?

C.3.2 Answers

Question	**What different kind of external resources are we dealing with?**
Answer	The external sources vary from airline companies sites to review sites to social networks. These all have their own way of inputting a review. For instance, Twitter does not have any format related to the review and only consists of lines text, while review sites already have a format in place that allows the user to rate some of the attributes (Food, timeliness, service etc.) on a given scale. The final application should be able to digest all these kinds of reviews and output the results on a scale . A separate , however still important, question relates to the quality of the data. The external sources may or may not have systems in place that assure the quality of a review.
Design Decisions	Dataformat, Text-to-rating

Question	**How is the data from different external sources combined?**
Answer	The sources have their own unique format as explained earlier in the document. If the system where to parse this into a general format information might get lost or lot's of information is undefined. However the general format helps decrease the complexity of analysing, because that system does not need to account for all the unique formats. This decision requires further collaboration with the data storage as that needs to be able to handle the data.
Design Decisions	Dataformat, Data Combining

Question	**How is the combined data used in order to obtain a rating?**

Answer	Not all external sources give a rating on a scale (e.g. Twitter) therefore these reviews need to be analysed first in order to be useful. There are multiple options available when analysing these results: 1. Drop all reviews that have no rating attached to them. This would lead to a huge data loss, but decrease the complexity of the system. The ratings can then easily be combined by computing a weighted average (or by another formula) of the individual ratings. 2. Transform the reviews without ratings by performing a sentiment analysis. After the analysis the ratings can be combined the same as in the other option.
Design Decisions	Text-to-rating

Question	**Is there any filtering process on the external data?**
Answer	Some reviews may not be considered useful, because of certain properties. The system could apply some business rules in order to filter those reviews out. The business rules governing these reviews are however currently unknown.
Design Decisions	Text-to-rating

C.4 Domain specific problems related to data storage and analysis

C.4.1 Import questions

This section lays out the domain related questions with data storage. In the earlier section analysis was inspected from a logical perspective. In this section the analysis is mainly inspected from a performance perspective. The system is responsible for storing several kinds of data both from the local system as well as data from external sources. This implies several questions:

1. How big is the data from external sources?
2. What kind of data does the system need to store?
3. How many databases do we need?
4. How do we handle large data-sets?
5. Will we always recalculate the ratings or will we combine the new rating with the known value?
6. Will we do incremental or time-interval updates?
7. Will the flight information be used to influence the rating?

The system has to store information such as users, reviews, statistics, data from external sources, etc. Because of the variety of data types it is important to understand these data types and how the system utilizes this data.

Question	**How big is the data from external sources?**

Answer	To get an idea of the amount of data we get from external sources, we use the following numbers. There currently exist over 600 airline companies. To calculate values for 600 companies one specific company (KLM) is taken and the values are extrapolated over the other airline companies: 1. Twitter: There are 1.4 Tweets per minute on KLM. This would indicate that there are approximately 600 * 1.4 * 60 * 24 = 1.008.000 tweets per day on airline companies 2. Facebook: There are 101277 talking about KLM on Facebook. This would indicate that in total 60.766.200 posts are about airline companies 3. Tripadvisor: There are currently 470 reviews on KLM. This indicates that there are 282.000 reviews on airline companies. The size of the data already shows that great care must be taken in the data-mining and analysis step regarding performance and scalability. As new external sources are added over time, this becomes of even greater importance.
Design Decisions	The data is considered Big Data

Question	**What kind of data does the system need to store?**
Answer	The data categories defined are: 1. Users: The system requires a user management system to log in and out on the web application. This user data needs to be stored persistently. An important quality attribute related to this user data is privacy, because it holds sensitive information. 2. Reviews: The users should be able to read and write reviews on the website. 3. Data from data-mining: The polling module (discussed previously) supplies the system with data from different external sources (e.g. Tripadvisor, Twitter). Because of the large amount of data scalability is a key quality attribute. 4. Analysis results: The data from external sources is analysed to form a rating and calculate statistics for the airline companies. Performance is a key quality attribute here as these results are shown to a potentially large number of end users, as well as scalability as the statistical data grows fast over time.
Design Decisions	The data is considered Big Data

C.4.2 Answers

Question	**How many databases do we need?**

Answer	The data categories defined are: The system has to handle data from multiple sources and different usage types. For instance, the data-mining and statistics require a lot of write commands as opposed to reads, whereas the main website data, such as reviews and users, requires more read commands as opposed to writes. For this reason the usage of specialized databases for each would increase performance and scalability. If more than one database is used than the data needs to be split up while still keeping the system maintainable. A this time we have several options: 1. Split up the main website data and data extracted from data-mining. 2. Split up the main website data, data-mining data and statistical data. 3. Split up the main website data and for each external source have a separate database containing the data-mined data.
Design Decisions	Data split-up

Question	**How do we handle large data-sets?**
Answer	The data collected from external sources require a great amount of storage over time. In addition to that, all the incoming data has to be analysed to create new ratings and statistical data. For this data both performance and scalability are the most important attributes. By utilizing an event type system the load can be balanced and only write data after it has been analysed, thus decreasing the amount of reads and writes the database has to run. The implication is that data in the event handler/queue is not persistent to the database and may be lost in system failures.
Design Decisions	How are large data-sets handled?

Question	**Will we always recalculate the ratings or will we combine the new rating with the known value?**
Answer	When new data is collected from external sources or when a review is made, the rating for the airline company needs to be updated. While a complete recalculation allows you to always change the weight of your external sources and reviews, it also means that over time more and more data has to be read from the database. Because of that the performance will deteriorate over time. To prevent this from happening, utilizing known values can greatly improve performance. By keeping track of the amount of tweets/reviews/etc received and the current rating, you only add the new input to those values by calculating the weight and the share it has on the total. By doing so you don't traverse the whole database thus greatly improving performance. If a recalculation is required this could always be initiated separately.
Design Decisions	Recalculate or combine rating?

Question	Will we do incremental or time-interval updates?
Answer	When new data is received, the rating and statistics need to be updated. This can either be done on a time-interval, such as once an hour, or using a queue and continuously update the data. To get the best performance and to safe on hardware cost, the latter is the most optimal solution. By spreading out the load, the machine does not need to handle high peaks and therefore you don't need to have a machine build to handle the peak when it happens. This means that the system required to run the updates can be run on a less powerful system thus saving cost. It also means that the data is always being updated and that the users get the most recent information possible.
Design Decisions	Incremental / interval update

Question	Will the flight information be used to influence the rating?
Answer	The stakeholder interest in flight-data was the decreasing of the weight of ratings during delays caused by things out of the hands of the company, such as bad weather, and to see a relation in the statistics between flight-data and the height of their rating. To ensure the integrity of the reviews and to improve the performance of the system, we advice against the use of flight-data to decrease the weight of reviews. By doing so we don't have to read from the database to get the flight-data to filter reviews thus saving resources and gaining performance. It also ensures the data integrity as it may be possible that flight-data is entered after bad reviews have already been added to the system. It is also the case that if a flight is delayed or cancelled the service from the airline company to accommodate the passengers is being put to the test. This should not be weighted less because the flight-data says the delay is due to bad weather.
Design Decisions	Flight Information

D Reflection on viewpoints

Behind each viewpoint there have been numerous modifications which lead to the current designs. In this appendix this process is reflected upon and also the current limitations of the designs are discussed. This add meaningful information to those reviewing the current architecture as well as to the architects themselves as it can point out flaws in the design process. The first paragraph shows the overall process for designing viewpoints while the second paragraph discusses a single view, the concurrency view.

The views were thought of as an extension, a graphical overview, of some important properties within the system. First a theme was discussed that described related concerns/decisions of importance . This usually lead to numerous concrete use cases were this theme was applicable. The use cases were further refined and how they were handled was graphically displayed in the view. One advantage of this more or less bottom up approach is that it supported specific use cases instead of hand-heaving arguments. The disadvantage was that most use cases did not generally fit into one overall theme which lead to overloaded views and also that some views might be more descriptive in a more creative, non-use-case design. An example is given in the next paragraph were the concurrency view is discussed.

In the concurrency view the architects tried to show the components that can work in parallel, but also the components that cannot work in parallel and form a single point of failure. The idea behind the view is to show how the performance and scalability requirements are satisfied. So numerous use cases were identified and how these were handled was shown in the view. However as pointed in one of the review sessions, the concurrency view is overloaded. The redundancies are part of the robustness and availability of the system not concurrency.

The overall learning process regarding viewpoints is that there are multiple ways to show a theme graphically. Besides the view the rationale is also very important as it captures why this design fits the solution.

References

[1] Bass et al. *Software Architecture in Practice*. Addison Wesley, Boston, 3rd Edition, 2012.

[2] Hayes-Roth, F *Architecture-Based Acquisition and Development of Software: Guidelines and Recommendations from the ARPA Domain-Specic Software Architecture (DSSA) Program*, Teknowledge Federal Systems, 1994

[3] IEEE Std 1471-2000, *Recommended Practice for Architecture Description of Software-Intensive Systems*, United States, 2000

[4] Lane, Thomas, *Studying Software Architecture Through Design Spaces and Rules*. Software Engineering Institute, Carnegie Mellon University, 1990.